

AFRL-RV-PS-
TR-2016-0159

AFRL-RV-PS-
TR-2016-0159

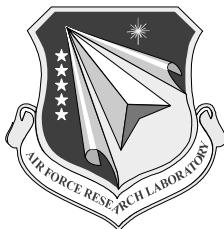
NANOSATELLITE AND PLUG-AND-PLAY ARCHITECTURE II (NAPA II)

James C. Lyke

28 Feb 2017

Interim Report

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.



**AIR FORCE RESEARCH LABORATORY
Space Vehicles Directorate
3550 Aberdeen Ave SE
AIR FORCE MATERIEL COMMAND
KIRTLAND AIR FORCE BASE, NM 87117-5776**

DTIC COPY
NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the PRS OPSEC Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (<http://www.dtic.mil>).

**AFRL-RV-PS-TR-2016-0159 HAS BEEN REVIEWED AND IS APPROVED FOR
PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.**

//SIGNED//
JIM LYKE
Program Manager

//SIGNED//
DAVID CARDIMONA
Technical Advisor, Space Based Advanced Sensing
and Protection

//SIGNED//
JOHN BEAUCHEMIN
Chief Engineer, Spacecraft Technology Division
Space Vehicles Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>	
<p>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</p>				
1. REPORT DATE (DD-MM-YYYY) 28-02-2017		2. REPORT TYPE Interim Report		3. DATES COVERED (From - To) 1 Dec 2014 - 1 Dec 2016
4. TITLE AND SUBTITLE Nanosatellite and Plug-and-Play Architecture II (NAPA II)			5a. CONTRACT NUMBER	
			5b. GRANT NUMBER	
			5c. PROGRAM ELEMENT NUMBER 62601F	
6. AUTHOR(S) James C. Lyke			5d. PROJECT NUMBER 4846	
			5e. TASK NUMBER PPM00015048	
			5f. WORK UNIT NUMBER EF126407	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Research Laboratory Space Vehicles Directorate 3550 Aberdeen Ave, SE Kirtland AFB, NM 87117-5776			8. PERFORMING ORGANIZATION REPORT NUMBER AFRL-RV-PS-TR-2016-0159	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/RVSW	
			11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited. (OPS-17-13046 dtd 1 Mar 2017).				
13. SUPPLEMENTARY NOTES				
14. ABSTRACT Under a second \$12M program agreement (PA) between US and Sweden (PA-TRDP-US-SW-AF-13-01), AFRL and Swedish Defence Material Administration (FMV) developed concepts from the previous nanosatellite and plug-and-play architecture (NAPA) program (e.g. a common "plug and play" technology) with four focus areas: (1) missions, (2) mission studies, (3) "i-Missions", and (4) technology development. The core "mission" activity involved development of a 6U-format Space Plug-and-play Architecture (SPA) Research Cubesat (SPARC). SPARC-1 (first and only pursued under this PA) demonstrates rapidly composable and service oriented spacecraft networks. In the mission studies focus area, the US/Sweden collaborative team explores concepts as diverse as mesh communication networks, synthetic aperture radar, combat search and rescue, blue force tracking, space situation awareness, and potentially other militarily relevant roles. The "i-Missions" focus area studies the kinetics of rapid mission development. The methodology involves interchangeable blackbox (self-describing) components, software (middleware and applications), advanced pushbutton tools supporting accelerated design flows, and elements of ground systems architecture capable of working fluidly with networks of potentially hundreds of these platforms. The technologies focus area develops miniature spacecraft components and subsystems. This program is on-going (projected complete in Spring 2020), and this interim report summarizes a snapshot, as of December 2016.				
15. SUBJECT TERMS Cubesat, nanosatellite, plug-and-play architecture, spacecraft, interfaces				
16. SECURITY CLASSIFICATION OF:		17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON James Lyke
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified	Unlimited	19b. TELEPHONE NUMBER (include area code)
50				

-- This Page Intentionally Left Blank ---

Table of Contents

List of Figures	ii
1.0 Summary	1
2.0 Introduction.....	2
3.0 Background.....	3
3.1 History of the NAPA Program.....	3
3.2 Program Vision.....	3
3.3 Organization of the Current Phase	4
4.0 Idealized Modular Cubesat	6
5.0 Mission Focus Area – SPA Research Cubesat 1	11
5.1 Mission Description	11
5.2 Spacecraft bus design	12
5.3 Payload design.....	19
5.3.1 SSA payload (PL2)	19
5.3.2 ASR payload (PL3).....	19
5.3.3 Additional experiment payload.....	20
5.4 Spacecraft integration.....	20
5.5 Ground architecture.....	20
6.0 iMissions	23
6.1 Motivation	23
6.2 Simplified Spacecraft Development Mission.....	26
6.3 Rapid Spacecraft (iMission) Ecosystem	28
6.3.1 Physical Modularity	28
6.3.2 Electrical / Interface Modularity.....	29
6.3.3 Software Modularity	30
6.3.4 Plug-and-play Infrastructure	31
6.3.5 How Plug-and-Play Supports Rapid Spacecraft Development.....	32
6.4 iMission Status	34
7.0 Program Status	35
8.0 Conclusions.....	38
References.....	39
List of Acronyms, Abbreviations, and Symbols.....	42

List of Figures

Figure 1: SPARC-X modular convention.....	7
Figure 2: Idealized plug-and-play wiring harness (blue wires are SPA-1 cables, black wires are SPA-S cables).....	9
Figure 3: Modular software.....	10
Figure 4: SPARC-1 mission overview.....	11
Figure 5: SPARC-1 top-level block diagram.....	13
Figure 6: Pumpkin SUPERNOVA	14
Figure 7: Data handling system (DHS).....	15
Figure 8: Agile Space Radio (ASR)	16
Figure 9: Hawk solar arrays	17
Figure 10: A broad view of the distributed software architecture throughout the SPARC-1 spacecraft.....	18
Figure 11: Cooperative morphing concept	20
Figure 12: Rendering of integrated SPARC-1.....	22
Figure 13: Steps in a Pushbutton ToolFlow (PBTF).....	26
Figure 14: SPARC-1 Physical Modularity.....	29
Figure 15: Summary of the iMission modularity.....	31
Figure 16: A shopping cart metaphor for rapid systems development.....	33

1.0 Summary

Under a second \$12M program agreement (PA) between US and Sweden, the US Air Force Research Laboratory (AFRL) and Swedish Defense Material Establishment (FMV) developed concepts from the previous Nanosatellite and Plug-and-play Architecture (NAPA) program (e.g., a common “plug and play” technology) with four focus areas: (1) missions, (2) mission studies, (3) “i-Missions”, and (4) technology development. The core “mission” activity involved development of a six unit (6U)-format Space Plug-and-play Architecture (SPA) Research Cubesat (SPARC). SPARC-1 (first and only pursued under this PA) demonstrates rapidly composable and service oriented spacecraft networks. In the mission studies focus area, the US/Sweden collaborative team explores concepts as diverse as mesh communication networks, synthetic aperture radar, combat search and rescue, blue force tracking, space situation awareness, and potentially other militarily relevant roles. The “i-Missions” focus area studies the kinetics of rapid mission development. Consistent with the plug-and-play model of the personal computer, the aspiration of the SPARC series (and the broader umbrella of research being done between the US and Sweden in the Nanosatellite and Plug-and-play Architecture or “NAPA” program) is to pioneer a methodology for creating mission capable 6U spacecraft. The methodology involves interchangeable blackbox (self-describing) components, software (middleware and applications), advanced pushbutton tools supporting accelerated design flows, and elements of ground systems architecture capable of working fluidly with networks of potentially hundreds of these platforms. The technologies focus area develops miniature spacecraft components and subsystems. This program is on-going (projected to complete Spring 2020), and this interim report summarizes a snapshot, as of December 2016.

2.0 Introduction

Space has always been important, and in the last decade it only seems more the case. Nanosatellites, which can be informally considered as small enough for a single person to easily lift, have captivated a lot of research attention. Their small size makes them less expensive to launch, especially the so-called “cubesats”, for which standardized canisters have been devised. As a result, hundreds of cubesats have been developed and launched.

Many things sets apart one cubesat from another: the technologies employed, the missions they are designed to carry out, the orbital parameters, and the ways they interact with users and operators. Most cubesats are technical curiosities, but some have been marshaled to powerful effect through scaled constellations (such as the Planet Labs “doves” [1]) that can deliver interesting services at global scale.

In this interim technical report, we describe the work to date in the joint development (by US and Sweden) within an international PA titled “Nanosatellite and Plug-and-play Architecture II” (NAPA II), approved 9 April 2013 as PA-TRDP-US-SW-AF-13-01. While the official title of the program is “NAPA II”, referring to the second PA between the US and Sweden, the program is often called “NAPA3” denoting the third research activity.

NAPA3 focused on development of a high-performance cubesat platform and architecture, referred to as “SPARC”, and the initial prototype (“SPARC-1”) was intended for launch in 2017 as a pathfinder carrying several experimental payloads. Its “value proposition” is based on a set of principles that include:

- Modularity in hardware and software, with ground support equipment and tools
- Avionics miniaturization
- Simplified ground architecture (“space dial tone”)
- Emphasis on missions based on finely-granular distributed constellations

This paper is organized as follows. In the next section we discuss the set of events leading to the present international program sponsoring the development of the SPARC architecture. We then discuss the idealized platform embodiment, followed by a more detailed description of the initial prototype (SPARC-1). We then discuss the current (as of the time of this writing) program status, and suggest the very promising landscape of what we believe may be possible with the SPARC platform in the future.

3.0 Background

In this section, we describe the evolution of events leading to a joint US/Sweden spacecraft development. The early interactions coincided with the evolving narrative of “responsive space” [2], driven by the sentiment that alternatives were needed to traditional acquisition approaches, which were sometimes characterized as costly, protracted procurement activities. The sentiment led to significant research investments (starting in the U.S. in 2004¹), as well as the creation of a DoD organization in 2007 dedicated to the principles of rapidly developing spacecraft to carry out missions in response to urgent needs [3]. The technological (vice political) viewpoint was in part that this was a “war on complexity”, leading to concepts that would make a vision of rapid spacecraft development possible. One concept, referred to as space plug-and-play avionics (SPA) [4], involved a combination of hardware, software, and protocol concepts that, in principle, when combined with the right tools and disciplines would allow spacecraft to be designed and put together rapidly, in mimicry of the plug-and-play mechanisms used in personal computers.

3.1 History of the NAPA Program

In 2006, Swedish researchers, in a US-sponsored “Windows on Science” visit, described progress in creating sophisticated nano-satellites that combined microelectronics, advanced packaging, micro-electromechanical systems (MEMS), and modular approaches. The work was striking in that it overlapped many of the same themes being pursued by the US Air Force in its own research programs. The modest meeting evolved into a decade-long collaboration between the two countries, building on a tradition of technology and architecture innovation.

NAPA is a three-part collaboration, informally initiated in 2006, between the Swedish defense material administration (FMV) and the US Air Force research laboratory (AFRL). The structure of the broader collaboration emphasizes the dual and complementary themes of miniaturization and modularity. It would involve an initial study phase to explore selected technology interests (emphasizing miniaturization in particular), followed by a second bilateral project to harmonize the modular concepts that had been separately under development by the two countries, to be demonstrated on minor spacecraft components. The third project most ambitiously would seek to develop entire spacecraft based on these concepts of miniaturization and modularity. So far, this plan has remained remarkably consistent with this nearly decade-long blueprint. An initial study project, commissioned by AFRL’s European Office of Aerospace Research and Development (EOARD) in 2007, implemented the intent of the first part of the collaboration, and it has been formally referred to as “NAPA1”. The second effort became a formal international agreement (2009-2011)², simply referred to as “NAPA”, but often referred to as “NAPA2” by our team. The third installment of NAPA became a second international agreement, spanning the period 2013-2020³, which is currently the active program whose work is described in the present paper.

3.2 Program Vision

NAPA was dedicated to the notion that nanosatellite platforms could be used to implement a number of mission roles having military utility. While clearly not capable of replacing all space

¹ Announcement from the first Space Plug-and-play Avionics (SPA) Workshop, Kirtland AFB, NM, July 13-14, 2004.

² Nanosatellite and Plug-and-play Architecture (NAPA), Bi-lateral Project Agreement between US and Sweden (PA-TRDP-US-SW-AF-09-002), May 2009.

Documents available by request to Secretary of the Air Force, International Affairs Office (SAF/IA)

³ Nanosatellite and Plug-and-play Architecture II (NAPA), Bi-lateral Project Agreement between US and Sweden (US-SW-AF-13-01), April 2013. Documents available by request to Secretary of the Air Force, International Affairs Office (SAF/IA)

missions, and limited in the size of sensing and energy capture apertures, these compact platforms could perform a number of useful tasks in communications, space and ground surveillance, and space weather. Furthermore, since the platforms are tiny, they can be proliferated. Many missions have a nature where “more is better” in terms of quantity. If we are tracking space debris, a single “eyeball in space” gives us a tiny amount of information. Thousands of eyeballs are better. Thousands of eyeballs with onboard processing and networking facilities are even better. Communication systems based on many tiny platforms, seamlessly networked together are capable of handling very large amounts of bandwidth in aggregate. Space weather sensors, even the simple ones that can be put on tiny platforms, can contribute valuable information. Here, too, more is better.

For the platforms themselves, many components could be miniaturized using advanced microelectronics, MEMS, and nanotechnologies. As we will discuss later, Moore’s law [5] and the technologies beyond it will further drive the useful amounts of computation that can be done in a given volume and a given amount of power. The flexibility of reconfigurable systems technologies, such as field programmable gate arrays (FPGAs) and software defined radios, add to the possibility of added utility, through adaptive features that can be reshaped even after spacecraft has been launched and placed into orbit.

Modularity and the ability to develop platforms rapidly contribute further to the potential value of a scaled fleet of nanospacecraft that can be designed, built, and deployed rapidly as opportunities are afforded or urgent needs materialize.

We suggest then that these ideas comprise the “NAPA construct” or value proposition:

- Nanosatellites can have military utility for certain roles
- Many of these roles benefit from scale (more is better)
- Technology helps (miniaturization, reconfigurability)
- The ability to move quickly provides more benefit.

3.3 Organization of the Current Phase

The current install of the NAPA program (NAPA3) is organized into a four track pursuit spanning technologies to missions.

Missions— This first track receives the most direct attention, as it involves the development of an experimental research spacecraft (SPARC-1), to which most of this paper is dedicated. It is the primary way in the present collaboration that we actually demonstrate technology elements in flight environment.

Mission Studies— The second track involves the examination of prospective mission concepts. In particular, we are interested in missions having military utility that can be based on massively proliferated constellations of experimental nanospacecraft exploiting the SPARC platform.

iMissions— The third track involves the study of modular architectures and their relationship to processes optimized for rapid design as well as development (assembly, integration, test) in

simulated on-orbit operation. It was clear that the primary activities comprising this track needed to be separated from the first track, since some of the tool concepts associated with iMissions are themselves under development and cannot be directly applied in any meaningful way to a project already underway to create an actual spacecraft. Rather, the work in this third track is a “clean sheet” activity which could be applied to some of the future mission concepts being pursued in the second track.

Technology Development— The fourth track simply refers to the eclectic bin of technologies pursued independently and jointly over the tenure of the NAPA collaboration.

4.0 Idealized Modular Cubesat

From the outset of the NAPA3 project, our team envisioned a modular cubesat architecture embracing the plug-and-play principles worked out in the previous phases of the NAPA collaboration. We established the 6U (10cm x 20cm x 30cm) cubesat as the foundation for mission developments, mostly for ease of launch with a canisterized satellite dispenser (CSD) [6] and the more capacious volume (compared to the more popular 1U-3U cubesats). We envisioned simplified design flows and ground architectures to complement the modular themes. In this section we discuss the idealized embodiment of the SPARC architecture.

Interoperable SPA-based Modularity— The plug-and-play model exemplified by the personal computer involves a set of conventions that power, data, control and configuration connections in components that plug into a host system. The conventions are so well-established that unskilled individuals can interchange components quickly and effectively. In our attempts to emulate the strategy for spacecraft, we adopted a set of conventions that became branded as “SPA”, which include:

- use of standard interfaces, including USB (SPA-U) [7], I2C based SPA standard interface (SPA-1) [8], Spacewire SPA standard interface (SPA-S) [9], and Ethernet (SPA-E), ideally with embedded power and standard connectors to permit connection through a single cable;
- use of self-description through extensible Transducer Electronic DataSheets (xTEDS) [10], embedded within components;
- use of self-organizing principles through protocol conventions that allow components to be discovered by the system when connected.

These we consider the core tenets of a SPA approach, complemented by a number of auxiliary concepts:

- flexible approaches to scale networks to larger sizes and bridge together subnetworks (through the use of routers, hubs, and adapters);
- standard middleware supporting an application structure mimicking the modularity interoperability hardware components (for example, having software applications employ xTEDS approaches for self-description);
- standardized modules designed to work with the standard, sometimes referred to as applique sensor interface modules (ASIMs) [11];
- testing hardware and software to establish compliance;
- tools to assist in the automatic generation of electronic data sheets and application code.

Structural modularity— A number of standards and support documents explicating the SPA concepts were published [12]. While these documents help in the creation of SPA systems, they neither constrained nor fully specified structural conventions. For 6U cubesats, Pumpkin Space Systems pioneered a new modular approach referred to as SUPERNOVA [13]. In the supernova system, a cannisterized satellite dispenser (CSD)-compliant substrate serves as an integrating surface onto which six 1U modules can be mounted in a planar (2 x 3) arrangement. The structural modularity complements the functional modularity of the SPA approach.

We studied extensions of this formulation, leading to further simplifications of this modular approach, which we refer to as the SPARC-X convention. The SPARC-X structural concept (Figure 1a) employs a pegboard like substrate onto which modular “building brick” components can be added (Figure 1b). These components (drawn from a prospective catalog of spacecraft components) are fastened to the substrate from underneath (Figure 1c). In the SPARC-X convention, components may take on any footprint of the form: $a \times b$, where a, b are in 0.25U (e.g., ~2.5cm) increments, but all are 1U (~100cm) in height, resulting in designs looking similar to those shown in Figure 1d. In effect, this theoretically reduces 3-D designs into 2-D “Tetris-like” layouts (though in practice it is not clear how many designs would cleanly fit into this convention). In this concept, the notion of the substrate as an integrating surface is reinforced mechanically, electrically, and thermally. To this end, the substrate would be relatively thick compared to the initial SUPERNOVA concept (Figure 1e), permitting the embedding of electrical cables (the wiring would be performed after modules were mechanically attached to the substrate). By convention, it would be expected that heat would be rejected into the substrate, which could be filled with a phase change material (PCM). The PCM would serve the dual purpose of assisting thermal management and providing structural support to the cabling.

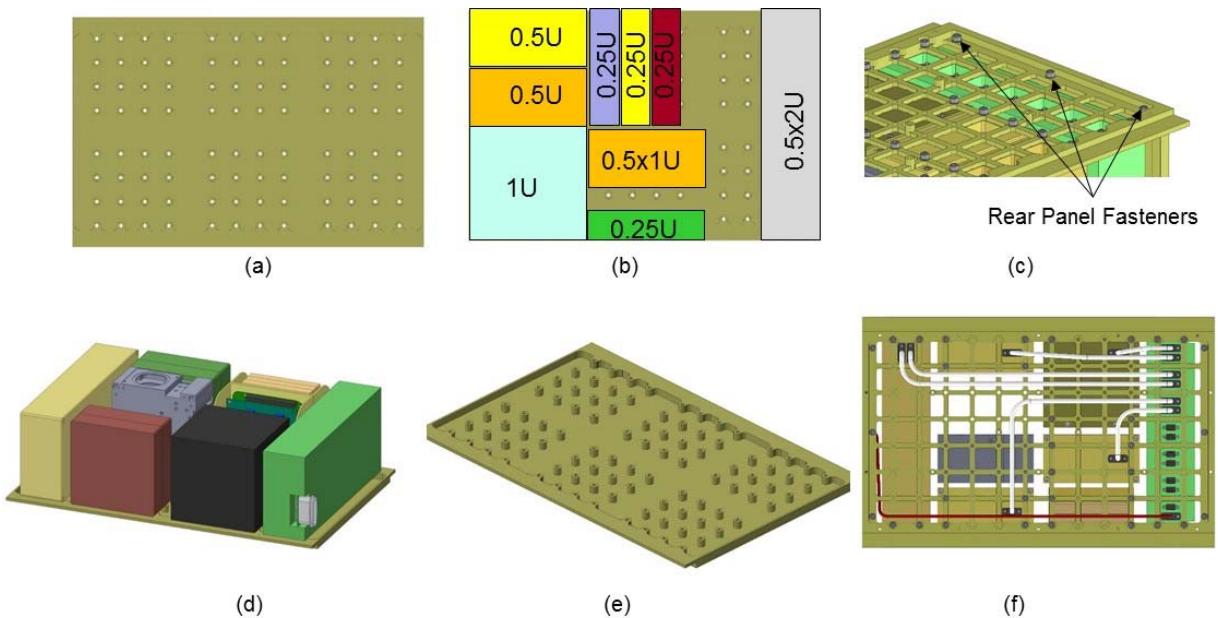


Figure 1. SPARC-X modular convention. (a) Pegboard substrate. (b) “Tetris” layout. (c) Mechanical attachment. (d) Example component layout. (e) Thicker substrate for wiring and thermal management. (f) Example wiring pattern.

Modular interfaces— The perfectly modular system would have single point interfaces between components. In the SPARC X concept, component would support either a high-speed or low-speed connection, represented as SPA-S or SPA-1, respectively. Earlier formulations called for integrating power into only the SPA-1 connection, which means all modules would support a

SPA-1 connection for power handling. Only modules needing a high-speed interface would contain the optional SPA-S connection. The difference in these approaches is shown in Figure 2 using an example spacecraft configuration containing a single command and data handling subsystem (CDH), and attitude determination control (ADAC) subsystem, the timing, telemetry, and control (TT&C) radio, an electrical power subsystem (EPS), and four payloads (PL1-PL4), three being high-speed (PL1-PL3) and one being low-speed (PL4). In the single point connection approach (Figure 2a), only seven cables are required to connect the eight components. In this case, the CDH contains a SPA-S router. As in the PnPSSat [13] approach, SPA-S connections are power bearing, and SPA-S routers also manage power distribution, leading to a simpler wiring harness but a more complex implementation. In the dual connection approach (Figure 2b), the EPS distributes all power connections and contains a SPA-1 signal router. High-speed connections receive an auxiliary high-speed (SPA-S) connection. Since this connection does not deliver power, it is possible to use traditional Spacewire cables, and it is not necessary for the CDH to manage power connections. The implementation is simpler, but has a more complex wiring harness (11 cables).

Idealized software architecture—SPA-based systems employ a layered software architecture that: (1) implements a “discovery and join” mechanism; (2) registers electronic data sheet contents in a lookup service; and (3) provides support for applications that exploit these features. Several of the plug-and-play frameworks, such as the SPA Services Manager (SSM), employ electronic data sheets in software application, leading to a uniform treatment of hardware and software in a plug-and-play system. These features are shown in Figure 3. The xTEDS electronic data sheet concept (Fig. 3a) provides an automated description of all accessible features of black box components in a semantically consistent and scalable way. Figure 3b depicts the plug-and-play layered model (network details are abstracted away). Figure 3b depicts the self-organization processes that occur through the use of specialized plug-and-play middleware (e.g., such as SSM).

Compliance testing for hardware and software—Our team has developed sophisticated ground support equipment that enable debug and testing. One of the tools, called virtual system integration (VSI), enables the remote connection between a host (e.g., CDH) in one physical location and the spacecraft component or payload in the second location. It is crudely analogous to the situation where an information technology help desk worker can remotely log in to a user’s computer to debug a software problem.

Simplified Ground Systems and Operational Architecture—One of the biggest barriers in spacecraft development is lack of consistency in standards in the design of communications architecture, which includes the spacecraft, ground receiving/commanding stations, operation centers, and connections to the user. As we began the NAPA-3 program, we were keenly interested in implementing the concept notionally referred to as “space dial tone” [14]. In this concept, spacecraft radio equipment (e.g., the “TT&C” blocks in Figure 2), like cellular telephones, would be designed to connect to particular networks. In a departure from traditional satellite communications design, each of the ground networks would interact with a unifying server system. Through careful protocol design, it would be possible to implement connections to the server by operators and users alike through mechanisms analogous to modern Web services. In principle, the SPARC-X platform, like a consumer purchasing the cell phone, would be free to choose from a variety of products and networks, and yet expect a consistent user

interface/user experience. Furthermore, it should be possible to create test radios that connect through ordinary wireless connections (e.g. 802.11) that approximate the experience of the spacecraft communications on orbit. Furthermore, it should be possible to mix-and-match several radios within the same platform, each ultimately connecting to the same unifying server, albeit through widely disparate paths.

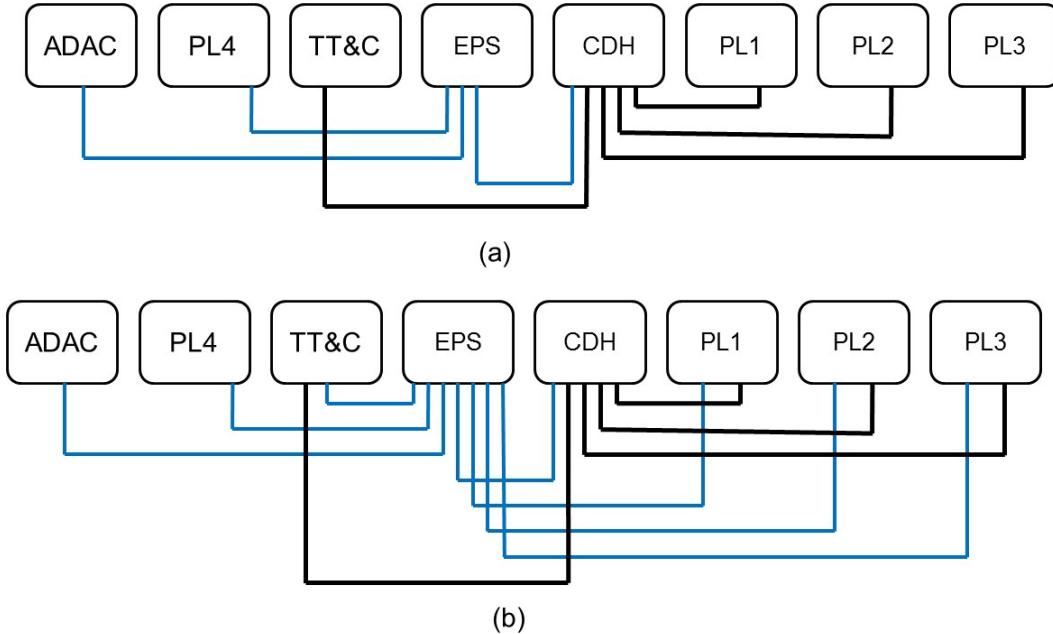


Figure 2. Idealized plug-and-play wiring harness (blue wires are SPA-1 cables, black wires are SPA-S cables). (a) Single point connections. (b) Dual connections (for high-speed modules).

Rapid design (push-button toolflow)—Even a plug-and-play system cannot design itself. In NAPA, we envision applying the metaphor of rapid computer design (through websites such as Dell) to spacecraft. We sometimes refer to this concept as the “push-button toolflow”, which would build upon approaches used in software configurators and integrated development environments used in electronic design automation. In the idealized case, a SPARC-X platform “recipe” could be designed, resulting in a complete specification of the buildable spacecraft.

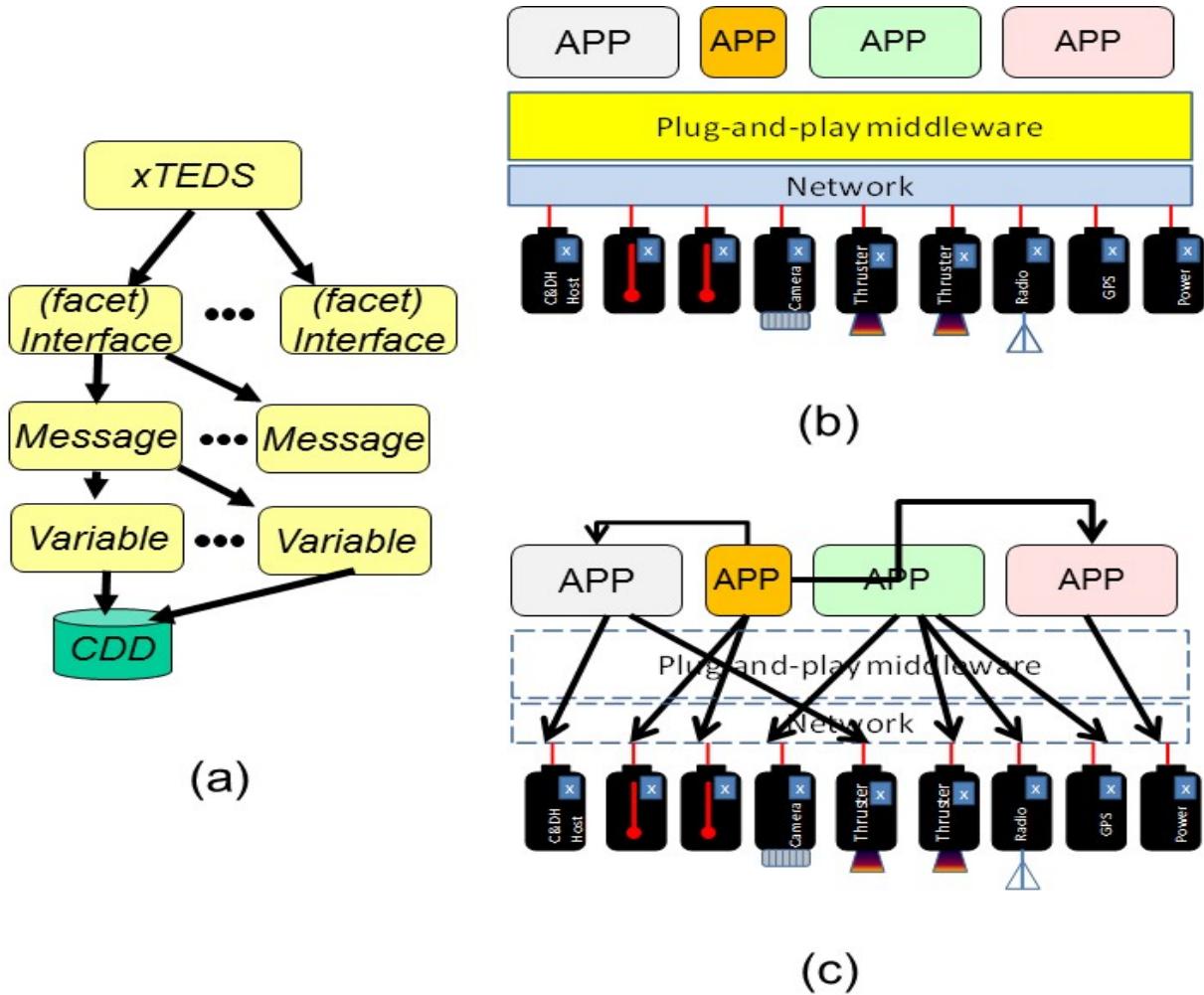


Figure 3. Modular software. (a) extensible transducer electronic datasheet data structure (xTEDS). (b) Plug-and-play vertical model. (c) Auto-organization.

5.0 Mission Focus Area – SPA Research Cubesat 1

There is the art of the possible, and then the reality of the practical. It is in this spirit that the NAPA team tackled the creation of a platform that could actually be developed and flown in the near-term that, while embracing many the principles of the idealized modular SPARC-X cubesat (described in the previous section), could not in fact implement all of the concepts in a time- and budget-constrained project. For Sweden, this spacecraft would be the first developed by its military organization. For the US, the spacecraft would serve as a proving ground for number of new technologies and architecture concepts.

In this section we discuss this “practical” spacecraft, called “SPARC-1”, which will carry several experimental payloads and demonstrate a high-performance bus architecture based on advanced avionics, a communication system with unprecedented flexibility, and modular structures.

5.1 Mission Description

The mission architecture is shown in Figure 4. SPARC-1 is a 6U cubesat to be launched in a low earth orbit (LEO) for up to one year (target date early 2017). In orbit, SPARC-1 will operate at least two experimental payloads. One payload is a space situational awareness (SSA) payload.

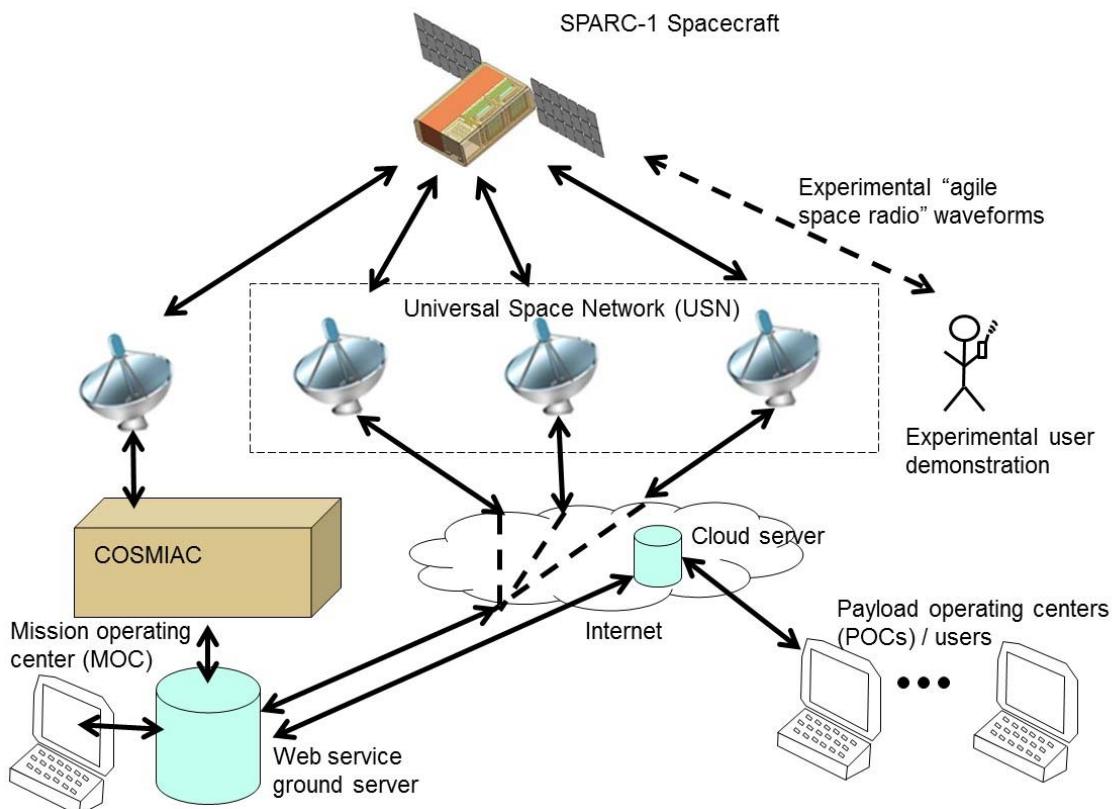


Figure 4. SPARC-1 mission overview.

A second payload is a flexible software defined radio, referred to as an “agile space radio” (ASR). A third experimental payload is under consideration at the time of this writing. SPARC-1 will conduct its primary (command and payload data) communications with several (we anticipate three) leased ground sites from the universal space network (USN) and a primary ground site at the Configurable Space Microsystems Innovations and Applications Center (COSMIAC) (Albuquerque, NM). SPARC-1 will also support interacting with ground demonstration users who can dynamically programmed ASR waveforms. A centralized ground server at COSMIAC will provide a universal interface for commanding the spacecraft, processing its data, and connecting to a cloud-based web service. Users can establish fixed or virtualized payload operating centers (POCs) to interact with this cloud system. Routine commands and payload requests are coordinated with the mission operating center (MOC), which plans and manages the mission.

5.2 Spacecraft bus design

SPARC-1 is a CSD compliant, 6U 30W three-axis stabilized spacecraft bus, having software defined S-band radio normally capable of 1 Mbps (and much higher under special conditions) downloads. It employs an OpenRisc fault-tolerant computer supporting multiple Spacewire (SpW) links (and other interfaces), running SPA middleware (SSM) on the real-time executive for military systems (RTEMS) [15] real-time operating system. It is capable of accommodating 2U-3U payload volume (currently about 2U is provisioned). The top-level block diagram based on the SPARC-1 preliminary design is shown in Figure 5.

Structural approach— While the advanced features in the Figure 1 structural approach were desirable, we knew the work necessary to evolve the architecture would unnecessarily add risk and complexity to our first spacecraft design. It was not a difficult decision to adopt SUPERNOVA as the baseline for SPARC-1, since Pumpkin worked closely with groups such as the US Air Force Institute of Technology (AFIT) [16] to test and mature the architecture, engineering the myriad of small details necessary to optimize it, freeing our team to focus efforts more on the contents of subsystems and less on debugging the elegant but immature structural conventions suggested in the Figure 1 system, which we continue to discuss with Pumpkin and explore in NAPA’s mission study and iMission tracks.

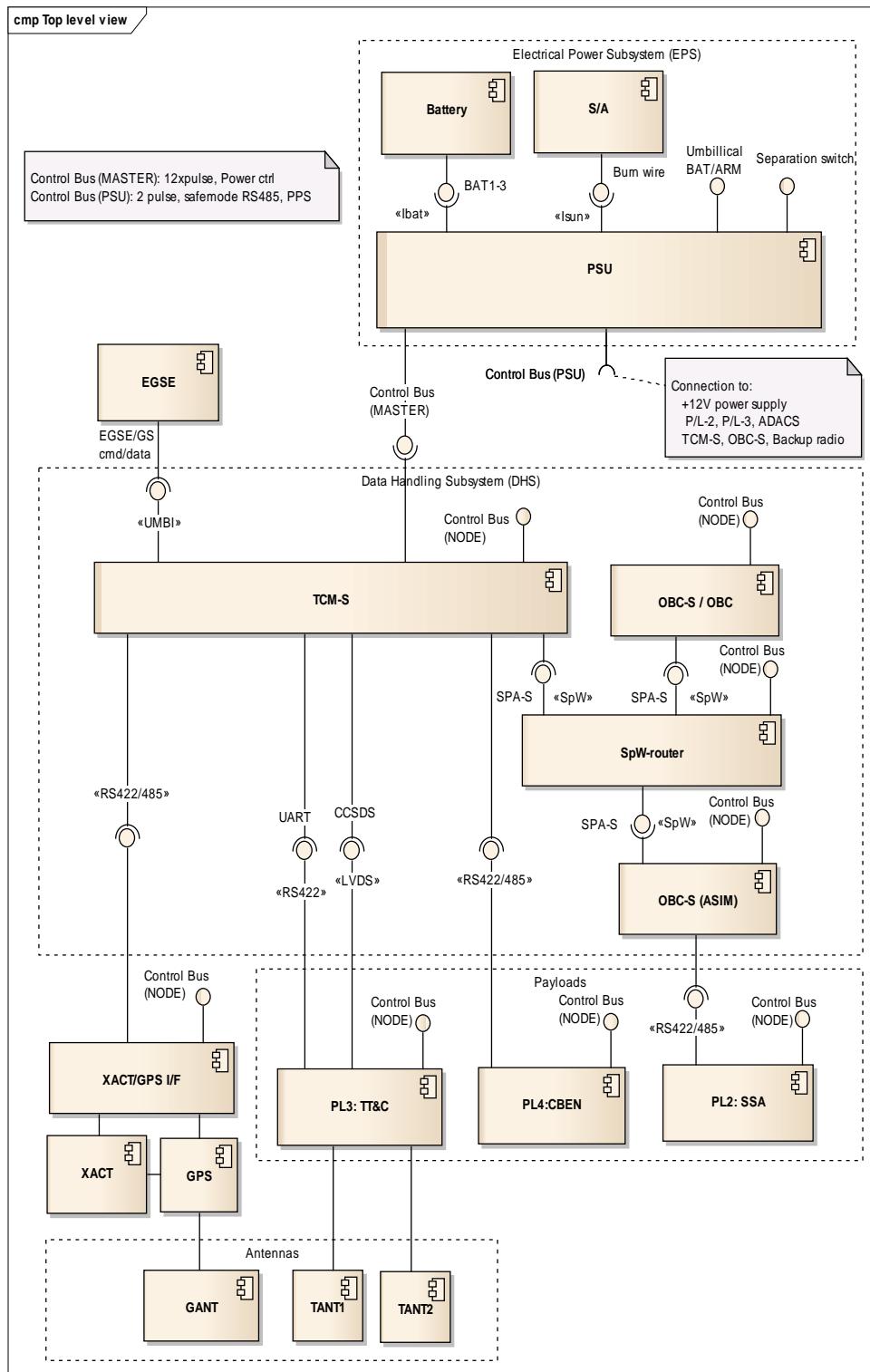


Figure 5. SPARC-1 top-level block diagram

The SUPERNOVA architecture is shown in Figure 6. The cage-like structure (Figure 6a) primarily involves a baseplate and a number of pieces (sides and top) that comprise a shroud like cover. There are six mounting zones, each accommodating a unit cell (Figure 6b). The unit cells are conveniently the same size as an entire 1U spacecraft (the structure kit sets for both are available from Pumpkin).

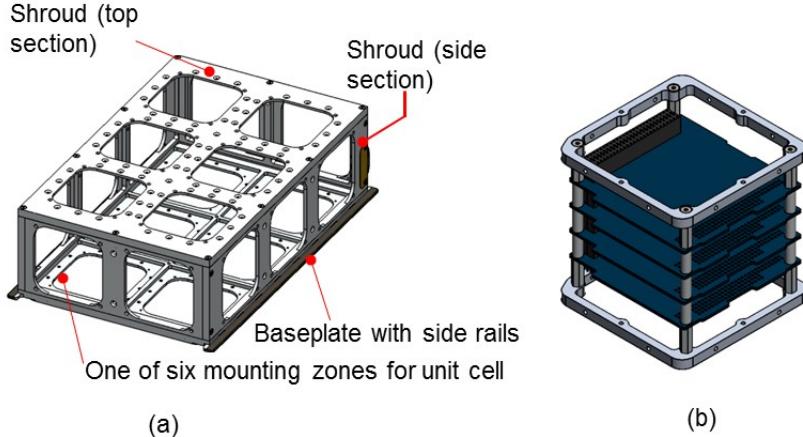


Figure 6. Pumpkin SUPERNOVA. (a) Structural cage and baseplate. (b) Example unit cell.

Avionics approach—The nexus of the SPARC-1 avionics facility is referred to as the data handling subsystem (DHS), comprising four-slice modular arrangement consisting of an onboard computer (OBC-S), telecommand module (TCM), a 6-port SpW router, and a second OBC-S serving as an ASIM interface to the SSA payload (PL2). The slice design, shown in Figure 7, fits into a cubesat envelope with recessed connectors, slightly thinner (20mm) than 0.25U module.

The OBC-S is the primary spacecraft central computer. It employs a 50 MHz OpenRISC [17] 32-bit processor instantiated in a Microsemi FPGA as a softcore intellectual property (IP) block, optimized for fault-tolerant and radiation-tolerant operation through triple modular redundancy (TMR) and error detection and correction (EDAC) approaches on the RAM (64 MB) in flash (1GB) memories. In addition to a 10Mbps SpW link, the OBC-S supports three universal asynchronous receiver transmitter (UART) (RS-422/RS-485) interfaces, inter-integrated circuit (I2C), and a variety of other support, test, and debug interfaces. The slice can be powered over a wide supply (5-16V) with a nominal power consumption of 1W.

The TCM is a mass memory system based on the OBC-S design, but more intimately interconnected with a 16 GB fault-tolerant mass memory store. TCM has direct support for Consultative Committee for Spacecraft Data Systems (CCSDS) protocols for telemetry [18-19] and telecommand [20-21], using state machines directly implemented as IP cores in hardware. A number of virtual channel (VC) buffers can be configured to queue various telemetry endpoints within the spacecraft. These are groomed into uniform CCSDS packet structure [22] (only the NRZ-L modulation is implemented [23]), which is sequenced for download through the spacecraft radio system.

The SpW router supports the remote memory access protocol (RMAP) and is built with open source cores developed by JAXA. It also supports TMR and employs EDAC and debug connectivity. It routes 10Mbps spacewire links and can support the data connection portions of the SPA-S [9] standard.

Attitude Determination and Control (ADAC) — After reviewing several other possibilities, the SPARC-1 is employing the Blue Canyon Technologies XACT attitude control system [24] developed through AFRL support, combined with a Novatel OEM 615 GPS receiver [25] and a separate GPS antenna (GANT). XACT represents an impressive engineering feat in its own right: a star tracker, three-axis reaction wheel set, three-axis torque rod set, magnetometer, inertial measurement unit, and sun sensor in 0.5U package. A small interface board will provide regulated power, signal conditioning, and connector translation to complete the interface to the spacecraft.



Figure 7. Data handling system (DHS). (a) Aluminum slice design. (b) Stack of four slices comprising the DHS.

It is a common practice in spacecraft design to distribute timing at a one pulse/second (PPS) interval. While the GPS receiver in the ADAC subsystem provides the signal, the DHS manages its integration into telemetry and distribution as needed throughout the spacecraft. The advantage to this approach is that in the event of a GPS failure, it is possible to provide a rudimentary backup, albeit far less accurate in the long term.

In the original design, SPA-1 was planned for the XACT interface. Instead, a "software ASIM" concept will be employed. The software ASIM is a software "object" designed to drive non-SPA interfaces (XACT employs RS-422, for example), but provides an API to the flight software that "serves" the expected xTEDS interface. In principle, a future "SPA-ready" XACT (or comparable ADAC system) could be directly plugged into the SPARC system, bypassing the software ASIM.

Communications subsystem— The ASR (Figure 8) is both the primary communications facility (TT&C) for SPARC-1 as well as an experimental payload (described later). By convention, the two functions are mutually exclusive. The TT&C function is the default mode, and any operations in “experiment mode” are terminated by a non-interruptible timeout mechanism after a nominal period (e.g., 20 minutes) to prevent lockout. In its TT&C mode, ASR is designed to operate with several ground stations (as shown in Figure 4). The functionality (Figure 8a) in TT&C mode involves a set of processing functions, including packet embedding within a frame structure, waveform encoding/forward error correction, encryption/decryption, etc. The ASR hardware, developed by Vulcan Wireless (Carlsbad, CA), is based on a decade-long evolution of compact software radio concepts that have been demonstrated on sounding rockets [26] and orbiting spacecraft [27]. For NAPA, this work was adapted to produce a flexibly configurable, compact (0.5U), and low power embodiment. To simplify integration into SPARC-1, the ASR and its two antenna will be arranged with mechanical adapters (Figure 8b) to conform to the SUPERNOVA unit cell conventions.

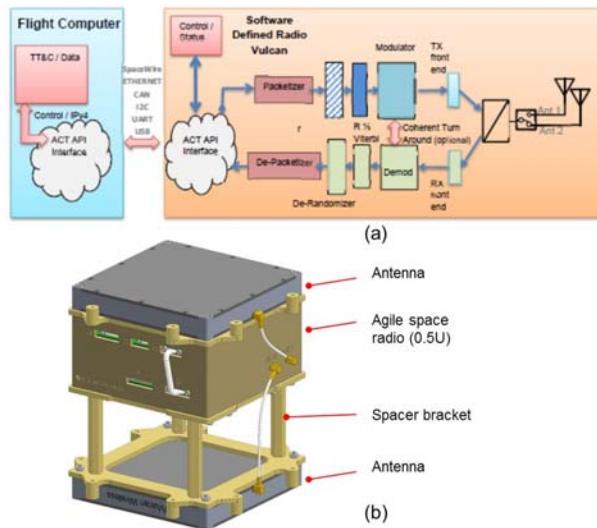


Figure 8. Agile Space Radio (ASR). (a) TT&C functionality. (b) Assembly arrangement as 1U cell.

Electrical Power subsystem— The electrical power system consists of: solar arrays for power generation, batteries for energy storage, and power management and distribution (PMAD) electronics. It produces about 20W average over an unregulated voltage span of 10-12.4V (in parallel with the battery banks and as mediated through charge regulation circuitry).

The “Hawk” solar array system (Figure 9) is being developed for NAPA through MMA Design (Boulder, CO). It is configured as two wings, each having three panels. The panels (each forming a string of seven series-connected solar cells) are folded and wings stowed by folding them at a hinged attachment point alongside the spacecraft in its launch configuration (Figure 9a). Once on orbit, a burn wire releases the wings (by command) to produce a deployed configuration shown in Figure 9b. The panel strings are parallel connected to eventually form the aggregate spacecraft power production,

One battery pack, configured with three cells in series and two in parallel (3S2P) will be used as energy storage, based on a standard product from GOMSpace (Aalborg, Denmark), with a capacity of about 58Wh.

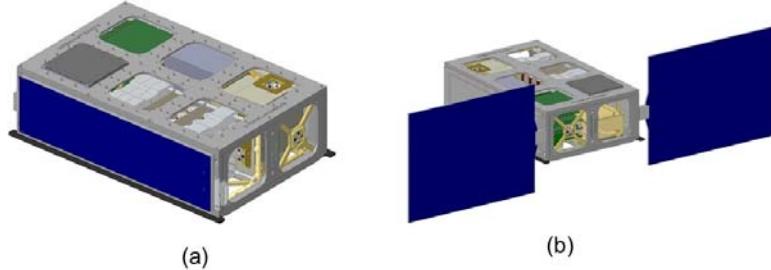


Figure 9. Hawk solar arrays. (a) Stowed configuration. (b) Deployed.

The PMAD electronics manage the regulation of charge between batteries and the solar arrays, instrument voltage and current, and distribute individually articulated power connections to the components of the spacecraft with two service levels. The "Class A" power taps provides access to the unregulated bus. While these taps have settable limits, they are considered to have "always on" connection, being active by default and turning on autonomously after a preset time in the event of an overcurrent event. By contrast, the "Class B" power taps are off by default, and must be commanded. These controlled power taps can draw power from either the unregulated bus or from a regulated 5V supply.

The EPS supports a number of safety and protection features. Blocking diodes and shunt protection are used to protect against faults involving individual strings. An undervoltage lockout switch is present which disables all parts of the spacecraft except for the battery charging in case the battery voltage would drop to a critically low level. The same switch is used to ensure that the spacecraft is disabled until a "remove before flight" interlock pin has been removed and the separation switch is triggered. Additional separation switches disconnect the battery from the spacecraft altogether during storage in the CSD prior to launch deployment.

Flight software— The organization of the SPARC-1 software architecture shown in Figure 10. Here, we show the distribution of software functions in the distributed modular system (i.e., the spacecraft contains processors in many different places).

The primary plug-and-play software is housed in the OBC. The architecture for flight software in the OBC largely follows the vertical layer model shown in Figure 3. Low-level drivers are used to implement the SPA-S interface and a software ASIM is used to virtualize access to the RMAP memory map in the spacewire router. The mission control suite of functions include scheduling; housekeeping (HK) / fault detection, isolation, and recovery (FDIR); and autonomous spacecraft (S/C) handlers.

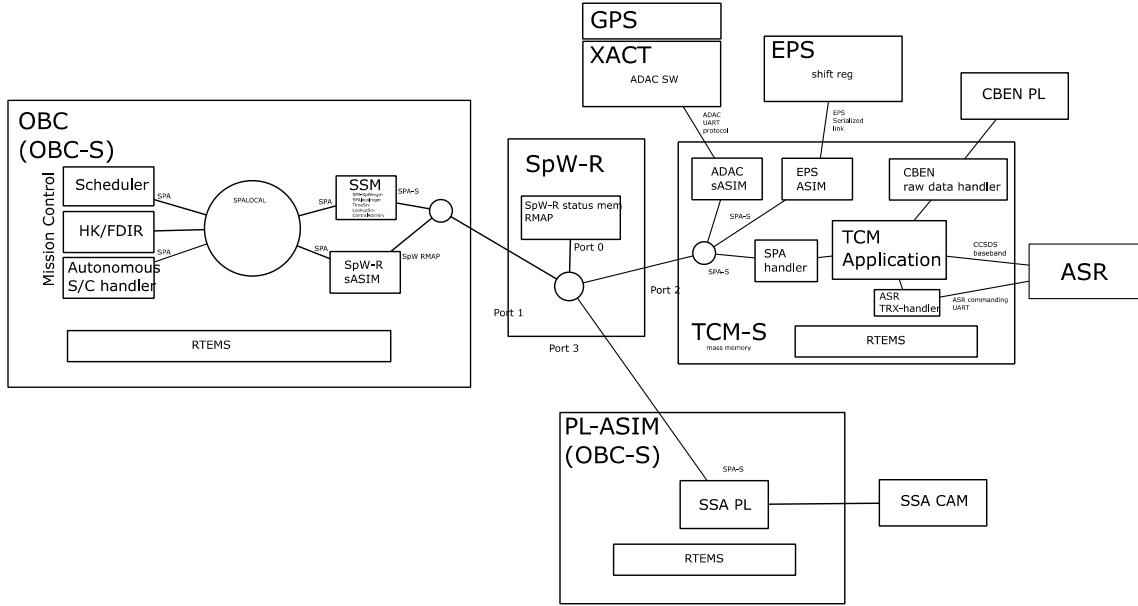


Figure 10. A broad view of the distributed software architecture throughout the SPARC-1 spacecraft.

Software in the TCM (also based on the RTEMS Operating System (OS)) does not implement SSM and SPA applications. However, it does interact very closely with the SPA model. For example, it employs SPA-S handlers that provide a "SPA wrapper" to access the TCM application code from the OBC, which itself convolves a number of functions that include the mass storage, telemetry queuing, and communications handling. It also hosts code to implement software ASIMs for the ADAC and EPS. Hence, the SPA model can be applied to powerful effect even in systems whose interior design does not apply SPA principles, but rather abstract them through the notion of "software ASIMs", which present the API in the form of a software accessible xTEDS.

The other primary processor in the DHS stack, the ASIM interface to SSA payload (PL2), also implement software on an OpenRisc processor using RTEMS. It contains the custom application code to service a camera and the SPA-S handler.

5.3 Payload design

Currently, SPARC-1 hosts two payloads, SSA and ASR (when in experimental mode). A third payload is under active consideration at the time of this writing.

5.3.1 SSA payload (PL2)

The purpose of the SSA/star-tracker-experiment is to gain knowledge that is going to support and increase the Swedish Armed Forces capacity of monitoring space over Swedish territory and during military operations abroad by the means of space situational awareness, SSA. The expected results from the SSA-experiment is to receive data from the slightly modified star-tracker with a quality that allows for detecting, localizing and possibly identifying objects in orbit. The primary science user is the Swedish Defense Research Agency (FOI) who processes the results. The results are reported to and support the Swedish Armed Forces.

The science objective expected to be met is to prove the concept of performing in-situ tracking of objects in orbit with a standard star-tracker that has been slightly modified for SSA-purpose. The science objective which may be met through continued accessibility by secondary investigators is to prove the concept of using star-trackers as hosted payloads as a tool that contributes to a capable global SSA-system.

5.3.2 ASR payload (PL3)

Figure 11 depicts the cooperative morphing concept in action. We assume in general it involves a desire to optimize some criteria, such as maximizing bandwidth to downlink information from the spacecraft to the ground. For purposes of simplicity, assume that we define a measure of "goodness" ($0 < r < 1$). Assume that we have two parameters $\{\lambda, \alpha\}$ that can be varied, and that there exists a pilot channel between the spacecraft on the ground which represents a reliable protocol to communicate results and next actions. An example experimental pass begins with the introduction of a hypothesis $\{\lambda_1, \alpha_1\}$ (in this case the spacecraft provides the hypothesis to the ground station). The hypothesis communicates (over the pilot channel) a particular setting of knobs that both sides implement. When they implement this hypothesis, the pilot channel is no longer accessible as both sides have cooperatively morphed in an attempt to confirm the hypothesis. Following this step, a predetermined sequence is transmitted in either simplex or duplex form, and statistics are gathered, resulting in the computation of r . After and agreed to interval has elapsed (the experimental timeline consists of a deterministic cycle of switches between experiment and pilot channel), the results are communicated over the pilot channel. The next hypothesis is generated, and the cycle is repeated. In this manner, the methodology is established both for morphing and for determining measures of merit. This schema can be exploited in many ways, against many types of criteria and against many types of "knob-turning" protocols. Cooperative morphing is only one type of experiment, which allows for the structuring of approaches to rapidly gather data in fielded experiments that might be used to more optimally design communications systems, such as those being used for combat search and rescue, blue force tracking, and many other purposes consistent with tag-tracking architectures. Other experiments can follow this work, to include working with new tag-tracking concepts in the

field, examining new concepts for internet distribution, pseudo-ranging, and radio-frequency metrology.

5.3.3 Additional experiment payload

The NAPA team is considering a technology demonstrator payload, referred to as the "combat search / blue force tracking enabler" (CBEN) cube, as a possible additional payload for SPARC-1. To simplify integration, the payload would largely be self-contained, receiving only a power connection and a simple serial link from the spacecraft. The cube would be based on a Pumpkin 1U cubesat kit, and would contain five small printed wiring boards, four containing individual sub-experiments, and the fifth containing a simple data handling system to collect data from the sub experiments. Each sub experiment would contain simple technology projects, such as a

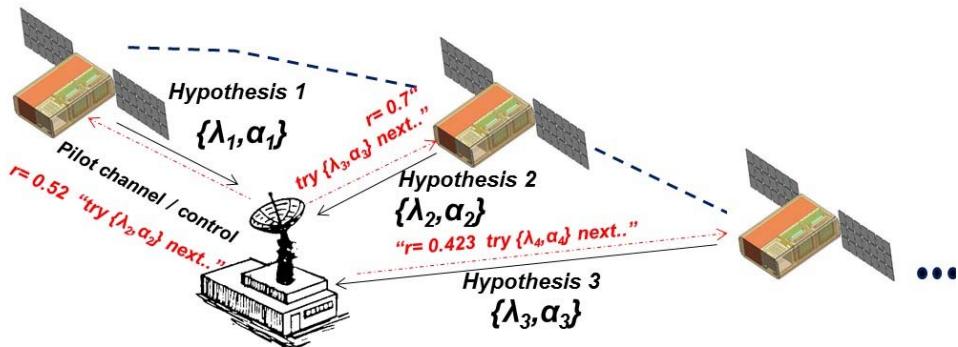


Figure 11. Cooperative morphing concept

MEMS switch, 3-D multichip module, or other devices. Given the tentative nature and the potential risk factors, participants would be cautioned that the project may not be manifested. The SPARC-1 architecture would be designed to accommodate its presence or absence without any first-order effects. If present, the payload would be scheduled using a best-efforts strategy, taking advantage of idle time not being exploited by other payloads.

5.4 Spacecraft integration

The primary work plan involves having the SPARC-1 spacecraft integration done in Sweden. Through the use of modularity principles, in mechanical hardware, electrical interfaces, and software, we hope to introduce a level of predictability that will simplify the task of integration. While the level of modularity is not as ambitious as described for a prospective SPARC-X reference architecture, the real-world SPARC-1 design represent a sensible blend of progressive modularity tempered by practicality and resource constraints.

Several views of the expected spacecraft configuration are shown in Figure 12.

5.5 Ground architecture

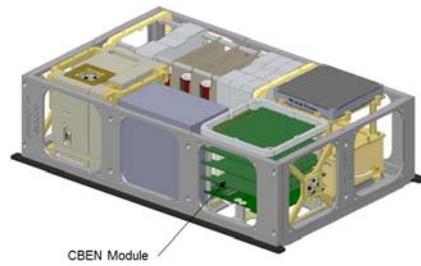
As suggested in Figure 4, the SPARC-1 ground architecture consists of several (three at the time of this writing) leased antenna sites, a central operating location (COSMIAC in Albuquerque), which is the mission operations center (MOC), and a collection of physical and cloud-based

servers that implement the interface between operators, users, and the spacecraft. The implementation problem can be roughly divided into two major sections. The first of these is the faithful replication of information being sent between spacecraft and ground (either commands going to the spacecraft or data received from the spacecraft). The second is the interface between particular users and the web system from which data is accumulated.

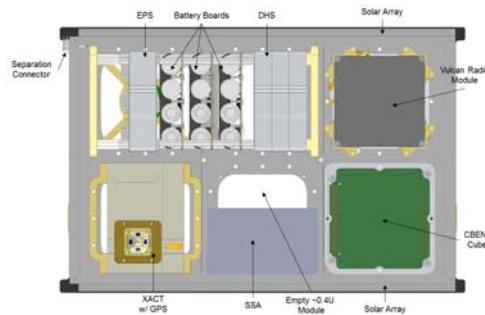
The problem of data transfer between ground and spacecraft is complicated by the fact that there are several pathways. Transactions may occur directly between the MOC site and the spacecraft when it is over the antenna co-located at the MOC site. Otherwise, the spacecraft is over one of the USN sites, which serves as a relay point. SPARC-1 will take advantage of a specially developed waveform, intended to be a preferred embodiment for small spacecraft that we refer to as simply the “open Internet standard (OIS)”. OIS is intended to simplify working from disparate relay locations. These locations are expected to intercept Radio Frequency (RF) energy from spacecraft, render them in a form that is easily transferable over the Internet, and then provide connections to a particular web portal. This can occur, even if data is encrypted aboard the spacecraft, since the relay sites merely recover a set of binary data (whether encrypted or not) and transports this data to an endpoint that presumably would have the keys necessary to decrypt and reconstruct the information, which in the case of SPARC-1 would be in the form of CCSDS packets. Both the MOC and the leased sites will support OIS.

The second problem then becomes the management of CCSDS packets and queueing of command requests. For this problem, we tend to implement a streamlined Web server architecture. At least in part, we intend to use cloud services, although the volume of traffic does not necessarily require the level of scale such systems could provide. It is a step in the direction of the “space dialtone” concept that we believe represents a better, more universal approach for users to interact with spacecraft.

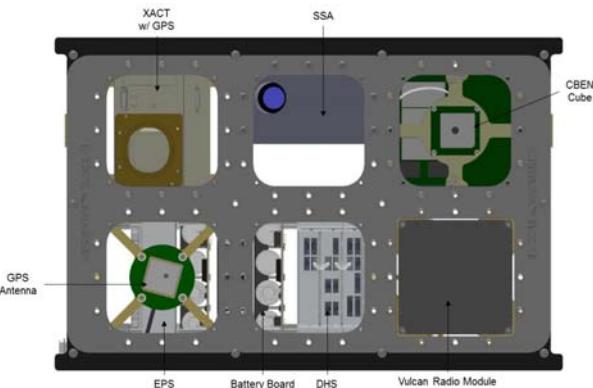
The ASR, when in experimental payload mode, poses other challenges to our hopes for unified ground architecture. When using nonstandard, special dynamically more formal waveforms, we cannot by definition conform to OIS. As such, we must operate over a specialty piece of equipment, and we are contemplating the use of GnuRadio [28] to support cooperative morphing. In other cases, we may configure ASR to implement legacy or experimental waveforms having no built-in mechanism to gather telemetry, meaning we must either collect the data separately (outside of the universal Web server), buffer experimental results in the ASR and seek a way to transmit them back into the spacecraft for subsequent downloading, or even simply buffer them inside the ASR and retransmit them in a way that is transparent to the rest of the spacecraft at some future point (considering that we may not actually be over the ground station when we conduct an experiment).



(a)



(b)



(c)

Figure 12. Rendering of integrated SPARC-1. (a) Perspective view. (b) Top view. (c) Bottom view.

6.0 iMissions

Acquisition programs are perennially challenged to more effectively translate user needs into effective material solutions [29]. In other words, developing systems takes too much time, costs too much money, and often features are eliminated to minimize overruns. Many attempts have been made to address the problem, and in the United States, the Operationally Responsive Space (ORS) program office was formed in 2007, partly in search for more efficient approaches. The ORS researchers examined many technologies, including modularity and plug-and-play (PnP) approaches [30], in hopes of improving efficiency by tackling some of the fundamental underpinnings that lead to cost and complexity, such as the discontinuity in interfaces leading to uncertainty in the processes of assembly, integration, and test.

In this section, we discuss work done on the specific problem of reducing the time to the degree possible in translating mission needs to generating orbiting and operational spacecraft. We suggest that while modularity and plug-and-play are helpful to rapid development (as these help eliminate uncertainty in hardware and software interfaces and protocols), other ideas are required, such as automation tools that manage the design and development. We describe an “iMission” approach that builds upon much of the work done by the US and Sweden in the previous decade in plug-and-play/modular spacecraft work.

6.1 Motivation

The time necessary to translate concepts into machinery that can be effectively mobilized to achieve a particular objective remains a central motivation in most human endeavors and especially in creating new capabilities for military systems. The ability to accelerate development usually is limited by one or more of the following causes:

- Thought-limited — It takes time to conceptualize, organize, plan, code, design, layout. This remains the most abstract and the fundamental problem in developing the system rapidly. Design processes represent a combination of rigorous modelling and analysis and ad hoc coordination amongst these processes
- Manufacturing Process-limited — It takes time to fabricate components, especially custom integrated circuits, printed wiring boards, wiring harnesses, specialty passive and active (deployable) structures.
- Geography-limited — Spatial distribution of elements, people, resources. It is necessary to bring many components from diverse locations to a single focal point of integration.
- Physics-limited — Time of flight limitations make it necessary to perform many processes on the spacecraft, such as rapid control loops.
- Coordination-limited — Time to communicate, refer, delegate, fill out paperwork, wait for approvals.
- Standards-limited — Lack of interchangeability and interoperability, which either forces re-engineering of interfaces or development of adapters and/or encapsulation concepts [31].

There are several trends that can deal with particular aspects of these time limiting factors, summarized in Table 1. Many of these trends are, by themselves, not complete answers. For example, the tendency to use central computers in spacecraft often results in a complex nexus of wiring, which triggers the need to produce a complex wiring harness. Decentralizing or even eliminating central computers would break up this nexus. Buildings pre-plan the provisioning of power, telephone, and computer (e.g., Ethernet) interfaces. While they do not eliminate the “wiring bundle” they relegate to almost casual insignificance. By contrast, creating a wiring harness for spacecraft is an exceptionally complex undertaking. The notion of simply moving to pre-planned power and (some) data distribution would result in potentially significant improvements. A more aggressive movement to modularity, standard plug-and-play, and even wireless interfaces would have the potential to even more dramatically simplify the wiring bundle.

It is also clear that the expense of radiation-hardened components has resulted in systems that will only cost more, but often have limited performance even compared to computers found in some consumer products. Developing specialty circuits, boards, and boxes based on such components further drives cost and complexity.

Software remains possibly the largest source of development and integration difficulty for almost any class of system. We indicate that several factors, including model driven design and other forms of automatic generation of code, can result in minimizing error-prone manual software development. Other methodologies and plug-and-play software development (middleware and the use of powerful and standard application programming interfaces) with the right disciplines can help keep these costs manageable by reducing fog and friction of interface uncertainty. The concepts of smart software frameworks, such as that in the so-called representational state transfer (REST) [32], and other software middleware systems, lead to effective modularity and composability approaches. This, combined with intelligence in component interface, are key principles in plug-and-play system development as they promote automated discovery and system organization when software and hardware components that comply with these principles were brought together.

Table 1. Current practice and future trends to improve rapid system development.

Current	Future
Central computers – More capable, more centralized, bigger wiring bundle	Elimination of central computers, distribution of intelligence in systems
Rad-hard parts – Captive processes, tremendous expense	Design-hardened, “good enough” radiation hardening, stronger leverage of commercial, much less expense
Wiring harnesses – Custom, cumbersome, expensive, long-lead time	Reduce / eliminate wiring bundles through wireless, pre-planned power distribution, and standard interfaces
Software – Complex customized development, tightly-coupled	Model-driven design, auto-generation of code, REST-inspired/plug-and-play software design
Reduced intelligence in components to reduce cost	Increased intelligence in components to reduce time
Methodical but non-integrated design flows with ad hoc automation	Push-button toolflows

But these concepts alone are not enough. Even the best plug-and-play systems, in which perfect composability exists, provide a situation little better than “monkeys on typewriters” (which given enough time, might theoretically create a literary masterpiece). In other words, there is no specific impulse to causes fragments -- even smart fragments -- to become a system. This is the central role of an overarching toolflow concept. We refer to this as a “push-button toolflow” (PBTF). Beyond “monkeys on typewriters”, PBTF provides an automated and guided workflow process, which navigates prospective users through a “wizard-like” process that translates imperatives into a buildable spacecraft. Beyond this, an effective PBTF approach would also coordinate launch opportunities for the spacecraft, communications infrastructure, and even manage coordination processes (i.e., “red tape” and paperwork) to the degree possible.

There is a precedent for workflow automation in complex systems, such as microelectronics design. For example, it is no longer necessary for designers to handcraft circuits and cut the shapes of transistors and wires from rubylith film (or even engage in the exercises of “pushing polygons” on a screen). Many of the design tasks are automated. In the era of the early microcontrollers (such as the Intel 4004), a team of designers would spend significant amounts of time to manually create simple circuits (by today’s standards) with a few thousand transistors, whereas today a small team can manage millions of transistors as “black boxes” (intellectual property or “IP” blocks), concentrating their efforts on integrating these modular blocks to form systems on a chip.

On the other hand, creating even a very simple spacecraft, such as a cubesat [33], currently involves many laborious steps, though below the complexity of a contemporary integrated circuit. In principle, we should be able to automate almost all aspects of routine spacecraft creation, from concept inception through on orbit operations. Were this possible, we believe the development of spacecraft could be made more predictable, less expensive, and quicker. This became the focus of the iMission research.

6.2 Simplified Spacecraft Development Mission

We briefly consider the “kinetics” of an optimistic and simplified spacecraft development, in part to understand how developments can be accelerated and why it is difficult to create systems quickly. We can view the process of spacecraft development for operation to include the steps shown in Figure 13. The view can be generalized to other platforms (i.e., not simply spacecraft).

This PBTF exercise assumes solutions can be implemented in the form of 6U cubesat spacecraft [16], consistent with the form factor chosen for the SPARC-1 development. The restriction is helpful in the study itself, and could be loosened to examine a broader class of platforms.

We envision this pushbutton tool flow (PBTF) to be guided by software wizards that help negotiate solutions. We imagine the flow to follow an open architecture that accommodates third-party plug-ins. Collectively, the software workflow system is comprised of a number of modeling, simulation, analysis, synthesis / configurator, and enterprise resource planning (ERP) functions. It is, in effect, a “platform build environment”.

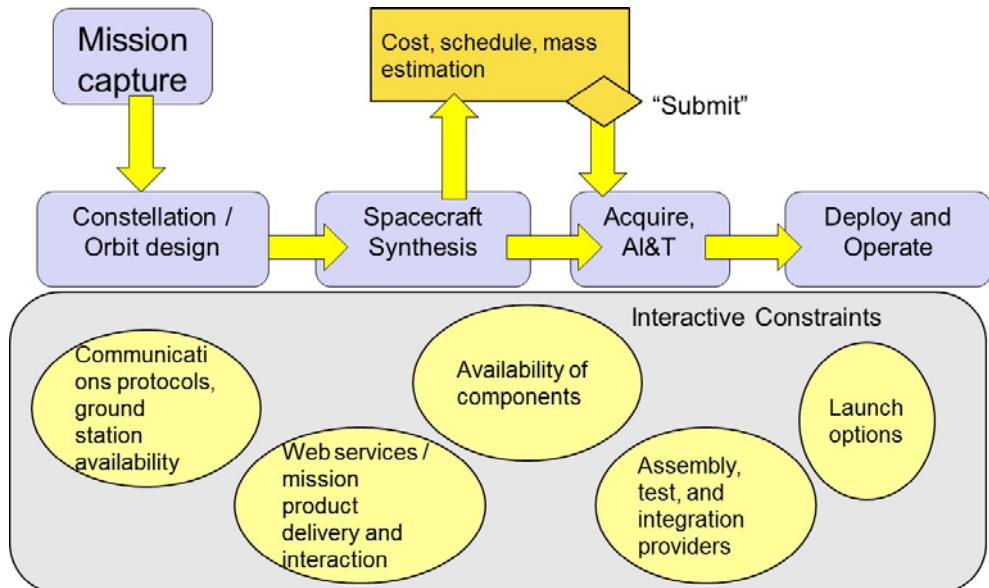


Figure 13. Steps in a Pushbutton ToolFlow (PBTF)

Mission capture. Mission capture refers to the process steps leading to encoding an unambiguous specification of what the user intends accomplish (for which spacecraft could be a possible solution) in a format that is conducive to analysis/synthesis. This amounts to a compressed mechanization of the requirements generation process. At this point, the user would engage with PBTF interactively to identify the broad categorization of mission (earth-observing, space-observing, *in situ* measurement, communications, special-purpose), and then activate decision aids that will extract information useful for determining a class of payload likely to satisfy the mission development. It is possible to adapt the flow according to whether payloads: (1) already exist, (2) do not exist (but can likely be obtained through requests for proposal), or (3) are “unobtainium” (partially vague, but having some minimal essential characteristics that permit the PBTF to operate and identify possible solutions). For example, we could consider a constellation of space environment monitors based on a still uncertain phenomenology but whose general global coverage and sampling intervals are known. Even without a concrete understanding of the exact type of sensor, there is enough information to begin to form a mission architecture with an “unobtainium” payload.

Constellation/orbit design. This step examines how to translate a user’s requirements into the orbital trajectories of one or more spacecraft that would as an ensemble satisfy the mission requirements. This step would also “comprehend” the intersection geometries for ground station coverage, time phasing of component operations (for example, in cases where payloads are only operated over particular points in the orbit and/or are over certain parts of the ground), and therefore begin to generate a “day-in-the-life” schedule, and generate desired launch times and insertion orbits. A more sophisticated version of the tool could analyze transfer orbits and determine propulsion necessary for final positioning. In effect, this stage of the PBTF derives many additional requirements from the mission capture phase.

Spacecraft synthesis. This phase of the pushbutton tool flow attempts to perform synthesis steps for each spacecraft in a constellation. Synthesis is a name given to processes used to compile software and digital hardware descriptions for FPGAs. Those cases involve translating high-level language descriptions into primitive machine instruction sequences or digital logic and memory elements, respectively. For spacecraft, the equivalent concept involves translating a basket of requirements and rules/constraints into a buildable spacecraft.

Examples of rules include:

- A spacecraft must have a source of power.
- A spacecraft must have an ability to communicate.
- A spacecraft (in the interim at least) must have a computer.

Examples of technical constraints include:

- A spacecraft mass (for a 6U spacecraft) should be < 12kg.
- Spacecraft volume must be < 6000 cubic centimeters.

- Temperature for electronics must not exceed 125 degrees C for more than 10 seconds.

Examples of user-imposed constraints include:

- Cost objectives below \$500K through launch and operations.
- On orbit in less than six months.
- Generated bandwidth < 1 Mb/day.

By employing physical modularity and plug-and-play mechanisms, as we will show in the next section, synthesis becomes a process of *selection*. By identifying an arrangement of components, after programming and configuration, that meet the requirements and satisfy the constraints, we have constructed the blueprint or recipe for a buildable spacecraft. This statement can be true even for components that *do not yet exist*, as we will discuss.

Acquire, assemble, integrate, and test (AAIT). The recipe that emerges from the synthesis part of the PBTF is a blueprint from which a spacecraft (or 100) can be built. In the AAIT portion of PBTF, the acquisition of components necessary for each copy of a recipe is coordinated. If we wish to build ten copies of a specific recipe, for example, we may find that eight guidance blocks can be acquired from one vendor, but two others must be acquired from the second vendor. Components can be tagged through software to key them for particular instantiations of a recipe, which is especially important when multiple copies of components are embedded in the same recipe design. This part of the tool most closely connects to vendor supply chains, so that component developers can register their offerings, third-party integrators can register their availability/fee schedule for performing integration services (or a user can choose a “DIY” or “do-it-yourself” option).

Deploy and operate. The “deploy and operate” phase of PBTF is concerned with the launching of all spacecraft for a particular constellation, preparing them for operation, and connecting them to users for command-and-control, as well as access to “mission products”. The PBTF presumes the use of Web services for much of this post-launch operation.

6.3 Rapid Spacecraft (iMission) Ecosystem

One approach to an ecosystem for rapid spacecraft development is to create a building block universe (physically and functionally) in which the participating elements must fit. For iMissions, we identify several levels of physical and functional modularity, along with other conventions for infrastructure that simplify the PBTF process. We believe this is may be the only appropriate application of plug-and-play conventions, to act in support of other concepts, such as PBTF.

6.3.1 Physical Modularity

In the case of the PBTF concept, we consider a modular physical LEGO-like system, which is a model used in modular “smartblock” phones [34] (for example, Google’s Project Ara [35]) and has been proposed for chemistry and biological instrument development [36].

The name of the modular block system for the iMissions is referred to as SPARC-X (Figure 14). It implements a pegboard-like “dinner tray” substrate (Figure 14a) onto which spacecraft components (depicted as blocks in Figures 14b-14f) are added. Early three-dimensional printed samples are shown in Figures 14g-14h. The SPARC-X physical modularity approach requires modules to have a 10cm height and have length and width fit a 0.25U (~2.5cm) grid. Hence, the blue module in Figure 14b is a 2U x 0.5U module, the grey module in Figure 14c is a 0.5U x 1U module, and the green module in Figure 14e is a 1U module. SPARC-X achieves physical composability by allowing any footprint combination of modules that fits over the “dinner tray” to be a potential spacecraft configuration.

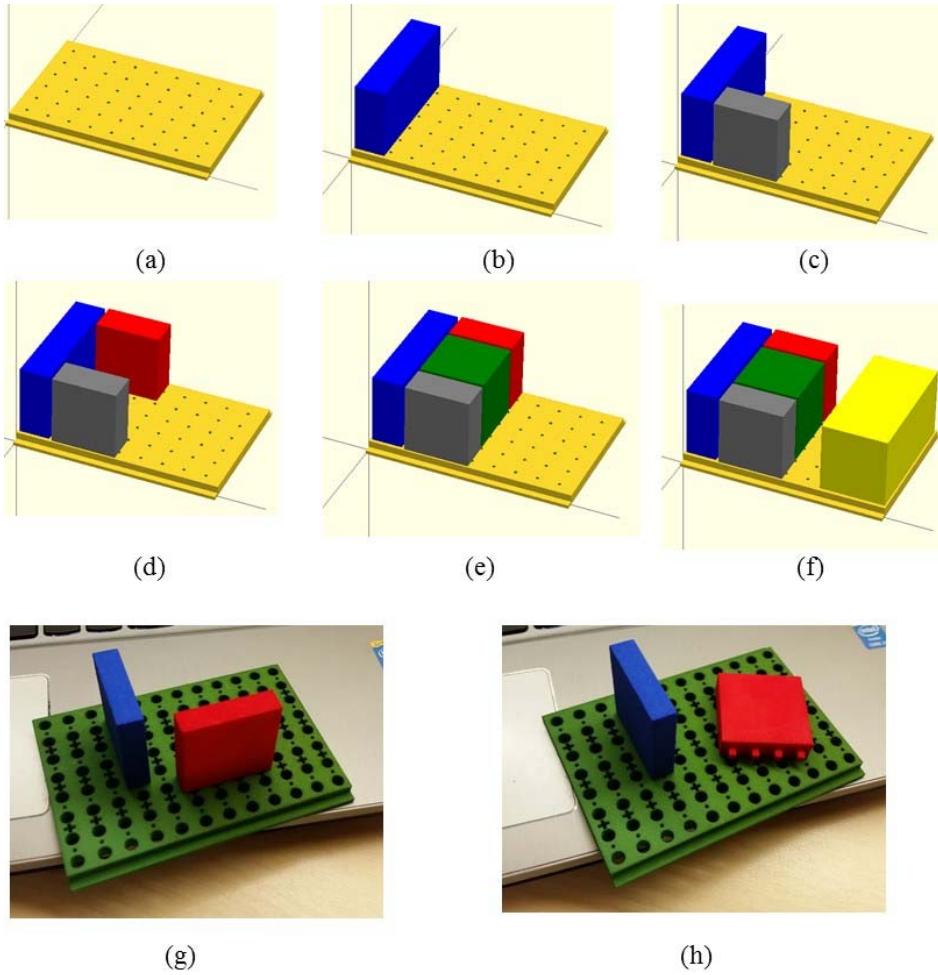


Figure 14. SPARC-X Physical Modularity

6.3.2 Electrical / Interface Modularity

The NAPA program has studied extensively the use of plug-and-play interfaces, such as those pioneered under the space plug-and-play architecture (SPA) program [37], also referred to as Modular Open Network ARCHitecture (MONARCH) [38]. For the iMissions, we identify three categories of electrical connection: (1) power; (2) data, command, control, and configuration;

Approved for public release, distribution unlimited.

and (3) custom. Power connections are handled through the substrate, which serves as a passive power distribution structure (similar to the wiring of outlets in buildings). In iMissions, data and commanding are handled through web interfaces. All modules act as web clients, except for the central computer, which acts as a web host. The modules employ wireless connections (802.11), eliminating the need for a physical wire interface. This approach allows for the near complete elimination of wiring harnesses in iMission spacecraft. To handle exceptional cases, such as routing an antenna with a coaxial cable, we identify category 3 to allow provisioning of custom cables. These custom cables can be tracked in the PBTF as separate pieces, part of the synthesis stage. For example, it may be necessary in order to implement a “type 301” radio, that two “type 407” antennas are necessary, and it is necessary to employ “type 53” cable assemblies in suitable lengths (as determined by the placement heuristics that define the arrangement of modules during synthesis). Specific vendors and lengths (part numbers) can be identified in advance, dramatically simplifying the harness problem.

6.3.3 Software Modularity

In the iMission concept, software is implemented in the same way as hardware, through the use of REST application programming interface (API) calls [39]. The primary mechanism in “RESTful” design is use of http: (web browsing) protocols. Hardware modules use wireless connections. Software modules can use scripted interfaces, employing mechanisms such as “curl” requests (these allow software procedures to mimic the actions of Web browsing clients). The concept is very powerful, in the sense that it is possible that a system comprised of 10 hardware modules and 30 software modules, each element can be in a different physical location (even a different city) and (excepting time of flight delays in custom cable connections) work as though they were on the same platform. Though REST-based designs have significant limits in real-time performance, they open new possibilities in distributed testability. For the iMission work in NAPA, where the “kinetics of the mission development are more important than the mission”, they provide an opportunity to explore far more aggressive concepts in rapid development.

The use of RESTful API in iMissions allows us to view the entire spacecraft recipe as a directed acyclic graph (similar to a tree) structure as suggested in Figure 15. The platform recipe amounts to a set of dependencies, some hardware and some software. In this approach, hardware modules act as wireless web clients. Software modules also act as web clients, but they can run as scripts (e.g. b1-b4 in Figure 15) within the central computer. In more complex systems having multiple computers, the software modules can be separated and executed on different computers (e.g., b1 in one computer, b2 in a second, etc.). Physical modularity allows blocks to be quickly put onto substrates, these providing a mechanical amount as well as a means of power delivery.

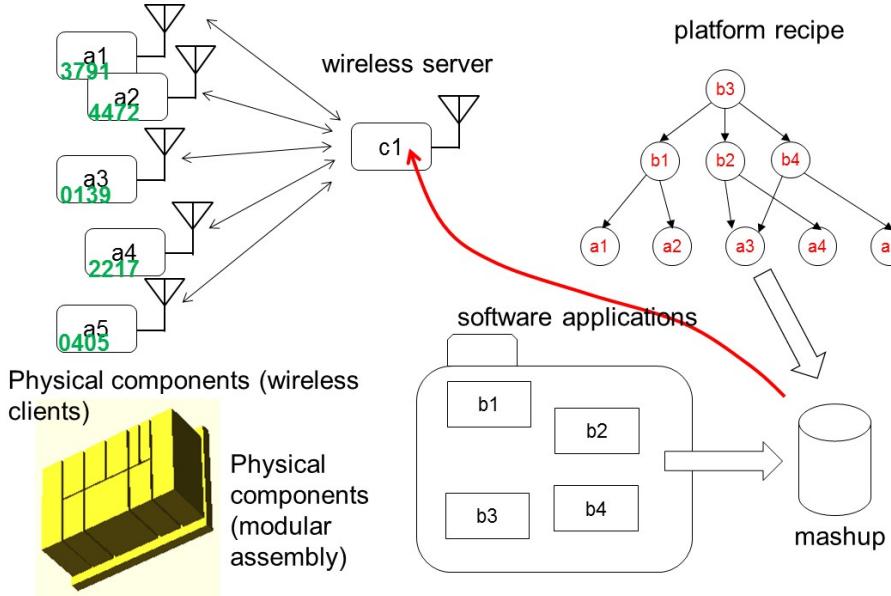


Figure 15. Summary of the iMission modularity/plug-and-play concepts for a single platform.

6.3.4 Plug-and-play Infrastructure

Beyond the spacecraft platform, it is necessary that the ecosystem include other infrastructure elements that embody some notions of plug-and-play operation. Without conditioning the elements to be PBTF-aware introduces significant impedance in the goals of reducing development timelines.

6.3.4.1 Launch

One of the most significant advantages of cubesats is their simplified connection to launch vehicles through the use of dispensers. For the 6U system in particular, a concept referred to as the canisterized satellite dispenser (CSD) is employed [6]. Any launch vehicle having spare lift capacity can be retrofitted with such dispensers. Doing so creates “rides” for spacecraft. In principle, a brokerage we might call “Launchworks” could be established, which (like airliners booking seats for passengers) could book rides for spacecraft. For iMission research, this fictional Launchworks concept eliminates uncertainty by providing mechanisms to automate the selection of launch opportunities for all the spacecraft in a particular constellation. Schedule and cost for the bookings, along with projected integration/certification fees, can be included, these allowing cost projections to be accumulated (along with launch schedule points) within the PBTF process.

6.3.4.2 Communications

One of the concepts discussed throughout the NAPA collaboration is the advent of an eventual “space dial tone” [14], which is a metaphor for simplifying spacecraft communications. Ideally,

prebuilt spacecraft radios (in the vernacular of iMissions, these would be plug-and-play building blocks using REST APIs to connect to the spacecraft central computer) would already be equipped and ready to connect to communication networks on ground or in space. The act of provisioning these radios for near immediate use would be as simple as provisioning a consumer cellular telephone purchased from a store. Significant progress is already being made in some architectures to make this vision possible, namely the use of Globalstar radios, which allow orbiting spacecraft to access communications through another orbiting satellite network. Data streams are billed by the byte and can be accessed through a web connection.

6.3.4.3 *Vendor Supply Chain*

In the early ORS concepts involving a satellite factory referred to as “Chileworks”, it was imagined that spacecraft would be organically designed and built within a single facility. The inventory would be managed and design processes analogous to the PBTF described in this paper would be used. It is possible to actually go well beyond this vision, such that the PBTF need not be operated inside of the factory, but anywhere that a web connection is available. By linking vendor networks through AAIT, it is not necessary to carry any inventory, nor is it strictly necessary to even touch a developing spacecraft. A real-world implementation of this concept would implement enterprise resource planning (ERP) concepts, so that vendors could register offerings into the PBTF dynamically.

6.3.5 How Plug-and-Play Supports Rapid Spacecraft Development

By eliminating interface uncertainty, it is not necessary to introduce additional physical structures and electrical circuits, which makes the footprints identified in spacecraft synthesis predictable and consistent. By adhering to standard electrical grids, modules receive power in a consistent way (analogous to the wall plugs in buildings). By employing software modularity concepts such as RESTful APIs, the uncertainty in software development is minimized since interface structure is standardized and PBTF can manage dependencies between components (whether hardware components or software functions).

When uncertainty can be adequately reduced, it is possible to begin to view system developments as being analogous to “shopping cart” exercises (Figure 16). In spacecraft synthesis and the acquisition portion of AAIT, we can clearly see the possibility of accumulating costs, generating critical paths leading to Gantt chart schedules, whether for one spacecraft or a dozen. We can choose to run automation for optimized analyses, in which we use tools (such as IBM CPLEX) that will implement decisions based on cost, schedule, and other encoded objective functions. Alternatively, we can allow users to hand pick components from catalogs, and they can witness the dynamic impacts to cost and schedule based on component selection.

Improving predictability in cost, schedule associated with mission development is one of the most important objectives in PBTF, perhaps as important or more important, than creating a spacecraft and the most rapid time possible. In some cases, we have ways of controlling how uncertainty can be allowed back into the process, through the use of “obtainium” and “unobtainium” constructs.

“Obtainium” constructs are those spacecraft components (hardware or software) that do not exist but are more or less obtainable with very high confidence. Examples of these include 3-D printed components [40] and auto-generated software [41]. These approaches lend themselves to situations where, for example in the case of brackets, it is not possible to stock in inventory a practically infinite number of variations, but a script can be used to generate the specifications (or in 3-D printing the .stl files) for a buildable element. It is possible to procure 3-D printed components (if not build in-house) through service bureaus (such as Shapeways [42]). Software can be parametrically generated in some cases through autocoding [43] and model-based systems engineering (MBSE) approaches [44-45]. In these “obtainium” cases, we have reasonable expectations of getting scheduling cost bounds for the associated components.

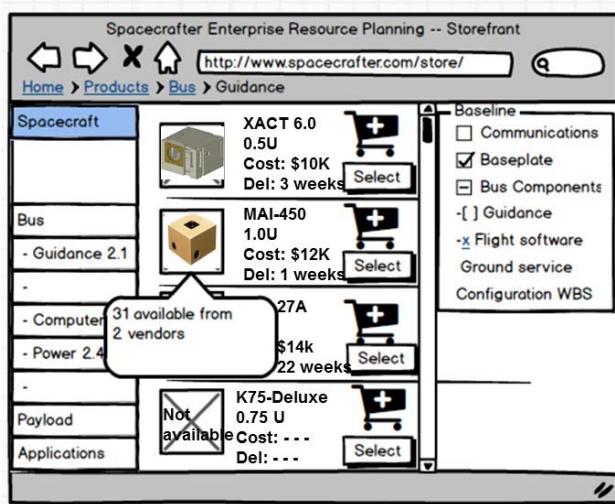


Figure 16. A shopping cart metaphor for rapid systems development.

“Unobtainium” constructs allow us to extend PBTF to include/manage uncertainty, but also therefore allowing risk to be added. For example, assume we need an imaging sensor that fits into a 2U envelope. We assign a placeholder component with estimated cost (e.g., \$500K) and schedule (e.g., 18 months) using special dialogue in the synthesis phase of PBTF. In doing this, we introduce engineering judgment (good or bad) alongside the more deterministic processes. We could then spawn an automatically generated request for proposal (RFP) that generates the specifications for this “unobtainium” component, updating our synthesis based on bids received. We could even conceivably allow a third party integrator to carry out the steps on our behalf. The power of doing this within the aforementioned ecosystem is that we eliminate several categories of uncertainty, to include the power, mechanical, and electrical interfaces, along with the expected software interface, which would be managed through REST APIs.

6.4 iMission Status

The work to create the iMission ecosystem is ongoing within the NAPA project. Several low-level studies have explored the use of IBM CPLEX to examine optimizations for cost and schedule using a fictional component catalog. Work on the REST APIs have shown that it is in principle feasible to set up a “closed cloud” server using a simple computer (such as a Raspberry Pi), with simple web clients using Wi-Fi-enabled microcontrollers. The use of REST APIs do not support high performance or determinism, but it is adequate for studies of rapid system development. We suggest that it is possible even in high-performance systems to employ REST APIs during initialization of a system with high-performance, since the REST API concept can be used to provision higher performance connections (e.g. web sockets or high-performance interfaces) between selected components. We expect within the NAPA collaboration to complete a rudimentary demonstration of the iMission ecosystem to provide insight into the degree of acceleration possible for creating simple missions based on 6U cubesats.

7.0 Program Status

At the time of this writing, SPARC-1 has completed its preliminary design, and (working through hundreds of actions identified by our team and independent reviewers) we are moving towards the critical design review milestone. Most engineering component deliveries occurred in 2016 (examples including the attitude control system, solar arrays, computers, and power subsystem unit). Mission studies have been approved to examine combat search/rescue, mesh communications, space situational awareness, and even synthetic aperture radar. The iMissions work has been examining the kinetics of rapid assembly, examining the SPARC-X platform as a basis for the modularity, push-button toolflow concepts. In addition to fluid interactions in the project, we support annual reviews for senior management involving general officer staff from US and Sweden.

Future Possibilities

We reflect briefly on the potentiality of the “NAPA construct” described earlier in this paper in light of the direction of future technological trends (such as the so-called “internet of things” [46]). It is for this reason that we comment that SPARC-1, exciting as it is, is merely the first step in a potential revolution in capabilities for proliferated networks of nanosatellite. In part these justify the optimism of the “concept car” exercises (SPARC-X and the iMissions) and they inform the mission studies which are a part of the NAPA collaboration.

Nanosatellites and the Emphasis on the 6U CubeSat — The original emphasis of NAPA (from 2006) was not on the cubesat platform, but rather on modularity and miniaturization. Several projects in Swedish spacecraft development around the time of the inception of NAPA were considered nanosatellites, but not in the form of cube satellites (cubesats), such as the Prisma [47] and Micro-Link [48] platforms. Over time, the popularity of the cubesat suggested a greater value might be gained in exploiting that platform for our mission concepts. The original CubeSat was a 10 cm cube, and it became the definition of a single unit (1U) standard. The advent of standardized launch containers, such as the poly-picosatellite orbital dispenser (PPOD) [49], made it possible to more economically launch several at a time. In particular, the PPOD can accommodate three individual 1U satellites within its tube-like container. It was only natural that some researchers sought to exploit the container to implement longer versions, resulting in 1.5U, 2U, and 3U sizes for cubesats. Later, the cannisterized satellite dispenser (CSD), which can be thought of as a “double-wide” PPOD, made it possible to accommodate a 6U cubesat. As NAPA entered its third collaborative phase, the Operationally Responsive Space (ORS) office emphasized the value of the 6U as having the greatest flexibility by virtue of the ease in accommodating these dispensers in a wide variety of platforms. The 6U, having a nearly 7 liter / 12 kg capacity, represented what the team felt was a critical mass for a variety of prospective missions.

The advantages of the 6U cubesat is the possibility of deploying large numbers of them (dozens to hundreds) from a single launch vehicle. Presently, they are parasitic “ride-alongs”, stowed as secondary / tertiary parts of a primary mission for which a launch vehicle was originally intended. It is conceivable that an entire launch vehicle dedicated to dispensers such as CSD could volley many more into orbit, allowing in principle the implementation of reasonably dense constellation. In the NAPA project, we consider a variety of missions that might be suitable for

such constellations. While our focus is on the “6U”, it is clear that larger platforms (such as 12U, 27U, etc.) could only improve the qualities of the missions we could implement. Hence, the existence proof for 6U is also an existence proof for satellites of any larger size.

Robustness of distributed systems— Examples of this existence proof involve a deeper consideration of missions that might be “fractionated”. Here, we refer to the possibility of breaking some missions that are done with large platforms into a set of smaller platforms that in combination approximate the qualities of the larger platform. Some missions, such as mesh communications, may be candidates for this philosophy. Networks of many small, egalitarian nodes follow the statistical dynamics of random networks, whose properties are distinct from those of hub and spoke (power law or scale free) networks. As commented by Albert Lazlo-Barabasi [50] power law networks are weaker when strong hubs fail, but random networks can operate under the loss of many individual nodes. As such, a carefully designed network of many nanosatellites may have a greater reliability than a smaller network of large spacecraft. This is very encouraging for space constellations that can be rendered in a disaggregated, proliferated form.

Moore's Law and the Landauer limit — While there has been much commentary on the perceived end of Moore's law, punctuated by the slowing trend of Complementary metal-oxide-semiconductor (CMOS) technology below 16nm in the consolidation of semiconductor facilities capable of reaching these refined levels of fabrication, there's room for continued optimism in micro miniaturization. We suggest that the revolution for 3-D packaging and 3-D integrated circuits is only beginning, and that there is a likely potential of many more orders of magnitude improvement in the coming decades. The Moore's Law potential for even the smallest satellites is staggering. Within two decades, we can expect a level of integration of one petabyte storage (50 libraries of Congress [51]) in cigarette pack form or (in nanosatellite terms) $\sim 0.25\text{U}$. Chip-scale atomic clocks (already presently available) can easily fit in the same form factor (in multiplicity, and combined with global navigation receivers capable of intercepting navigation signals from a growing variety of orbiting platforms). Dynamically-configurable wideband agile space radios (generations beyond the ASR in SPARC-1) and low-power inter-satellite (laser possibly) crosslinks will enable self-scaling mobile communication links (each also $\sim 0.25\text{U}$). Some may question ambitions of extensive on-orbit processing, but in this case the laws of physics offer hope. Decades ago, Rolf Landauer [52] identified the energy bound for manipulating a single bit of digital information as $O(kT \ln 2)$, a limit many orders of magnitude below that of contemporary CMOS. If processing, even on a nanosatellite power budget, can be improved one-million-fold, it may be simpler to have these systems operate as simpler cloud centers, where such future “fog computers” (i.e., distributed clouds) are comparable to a contemporary terrestrially-based data center. This scale of processing suggests disruptive possibilities, such as leaving data on spacecraft instead of shuttling information around the globe for processing in a fixed data center. Also, we can store vast knowledge repositories in these miniature orbiting systems, leaving the information there unless needed (to include the entire mission life event histories of these platforms).

Aperture, aperture — As we discussed before, the apertures (for solar, antenna) are the primary limits for the intrinsic mission capabilities of the spacecraft. While we do not anticipate breaking the laws of physics, we can expect far more creative work as possible in deployable structures and materials research. The prospect of *in situ* additive manufacturing may seem fanciful, but

would not appear to require breaking known laws of physics (but a matter "merely engineering" protocols and mechanisms to permit, for example the direct implementation of a portable fused deposited material 3D printer within an orbiting platform). While well beyond the current abilities of the NAPA program, we may possibly expect even in our lifetimes to see spacecraft fabricating elaborate structures to extend their effective apertures, using techniques far too frail to implement on the ground, much less survive launch vibration.

8.0 Conclusions

In this report, we have discussed a promising body of work being done as part of a collaboration between the US and Sweden in the NAPA program. Most of this paper has focused on the most visible part of the project, which is the development of the SPARC-1 spacecraft. It represents a very capable nanosatellite platform, one potentially far more capable than we envisioned at the beginning of our collaboration in 2006.

In some respects, the more important work may be in the shaping of ideas and visions for the future in which nano satellites play important contributions to space capabilities. While we have at several points stressed that the satellites do not replace all platforms, we believe there will be a number of important mission roles that they will satisfy as well as much larger platforms do today. In that respect we believe the work in this collaboration is groundbreaking, as our value proposition lies in the combination of technologies, architectures, and in reinforcing the trends that are already beginning in industry to examine and implement missions based on proliferated networks of nanosatellites.

REFERENCES

- [1] "Planet Labs," Wikipedia article, https://en.wikipedia.org/wiki/Planet_Labs retrieved January 11, 2016.
- [2] Wegner, P. and Kiziah, R., "Pulling the Pieces Together at AFRL – Space Vehicles Directorate," Proceedings of the 4th AIAA Responsive Space Conference, Los Angeles, CA, 24-27 April 2006.
- [3] "Operationally Responsive Space," Wikipedia article, https://en.wikipedia.org/wiki/Operationally_Responsive_Space_Office, retrieved January 11, 2016.
- [4] Lyke, James; Young, Quinn; Christensen, Jacob; Anderson, David, "Lessons Learned: Our Decade in Plug-and-play for Spacecraft," Proceedings of the 28th Annual AIAA Conference on Small Satellites, Logan, UT, 2014.
- [5] Gordon E. Moore, "Cramming more components onto integrated circuits," Electronics, pp. 114-117, April 19, 1965.
- [6] "Cannisterized Satellite Dispenser," Data Sheet, August 3, 2015, Planetary Systems Corporation, Silver Spring MD, <http://www.planetarysystemscorp.com>.
- [7] Lyke, J., Cannon, S., Fronterhouse, D., Lanza, D., and Byers, T., "A Plug-and-play System for Spacecraft Components Based on the USB Standard," proceedings of the 19th Annual AIAA/USU Conference on Small Satellites, Logan, UT, 8-11 August, 2005.
- [8] Lyke, James; Mee, Jesse; Bruhn, Fredrik; Chosson, Gael; Lindegren, Robert; Lofgren, Henrik; Schulte, Jan; Cannon, Scott; Christensen, Jacob; Hansen, Bryan; Vick, Robert; Vera, Alonzo; Calixte-Rosengren, Josette, "A Plug-and-play Approach Based on the I2C Standard," Proceedings of the 24th Annual Conference on Small Satellites, Logan, UT, 8-11 August 2010.
- [9] P. McGuirk, G. Rakow, C. Kimmery, P. Jaffe, "SpaceWire Plug-and-Play (PnP)," Proceedings of the AIAA Infotech Conference, 7-9 May 2007, Rohnert Park, CA.
- [10] Lanza, D., Vick, R. and Lyke, J., "The Space Plug- and- Play Avionics Common Data Dictionary - - Constructing the Language of SPA," Proceedings of the 2010 AIAA Infotech Conference, 20-22 Apr 2010, Atlanta, GA.
- [11] Lyke, James, "Plug-and-play as an Enabler for Future Systems," proceedings of the AIAA Space Conference, September 2010, Anaheim, CA.
- [12] Space Plug and Play Architecture, AIAA Standards available from <http://arc.aiaa.org/series/aiaastandards>, accessed June 1, 2014.
- [13] "Building a Plug-and-play Satellite," video, IEEE Spectrum, hosted on the YouTube website (https://www.youtube.com/watch?v=8UQ_ghEGD14, accessed June 1, 2014).
- [14] Ryan L. Buffington, Craig J. Kief, R. Scott Erwin, Joshua F. Androlewicz, and James Lyke, "GENSO, SPA, SDR, and GNU Radio: The Pathway Ahead for Space Dial Tone," Proceedings of the AIAA Infotech Conference, 29-31 March 2011, St Louis, MO.
- [15] Real-Time Executive for Multiprocessor Systems, <http://www.rtems.org>. n.d.
- [16] Latta, Robert C., "Structural Analysis of a 6U Cubesat Chassis," M.S. thesis, Dept. Aeronautics & Astronautics, AFIT, Wright-Patterson Air Force Base, OH, 2014.
- [17] "OpenRISC Project Overview," <http://openrisc.io/>. n.d.
- [18] "TM Space Data Link Protocol," Blue Book, Issue 2, September 2015, CCSDS 132.0-B-1, available from <http://public.ccsds.org/publications/BlueBooks.aspx>.

[19] "TM Synchronization and Channel Coding," Blue Book, Issue 2, August 2011, CCSDS 131.0-B-2, available from <http://public.ccsds.org/publications/BlueBooks.aspx>.

[20] "TC Synchronization and Channel Coding," Blue Book, Issue 2, September 2010, CCSDS 232.0-B-2, available from <http://public.ccsds.org/publications/BlueBooks.aspx>.

[21] "TC Synchronization and Channel Coding," Blue Book, Issue 2, September 2010, CCSDS 231.0-B-2, available from <http://public.ccsds.org/publications/BlueBooks.aspx>.

[22] "Space Packet Protocol, Blue Book, Issue 1, September 2003, CCSDS 133.0-B-1, available from <http://public.ccsds.org/publications/BlueBooks.aspx>.

[23] "Radio Frequency and Modulation Systems-Part 1: Earth Stations and Spacecraft," CCSDS 401.0-B-25, Blue Book, Issue 25, February 2015, <http://public.ccsds.org/publications/BlueBooks.aspx>.

[24] "XACT High-Performance Attitude Determination for CubeSats," product flyer, Blue Canyon Technologies, Boulder, CO, <http://bluecanyontech.com>, n.d.

[25] "Receivers OEM615: Compact, Dual-Frequency Gnss Receiver Delivers Robust RTK Functionality," product flyer, Novatel, Alberta, Canada, <http://www.novatel.com>, n.d.

[26] Lynaugh, Kevin J., Scott Dossey, Matthew Davis, Richard Middlestead, James Lyke, James Crane, John Demello, and Calvin Roman, "SPA-Based Emergency Micro Black Box Transponder for Space Vehicles," Proceedings of the AIAA Infotech Conference, 19-31 March 2011, St Louis MO.

[27] David, Carrie, "Team tracks recently launched ORSES satellite," Army.mil website new announcement, http://www.army.mil/article/115784/Team_tracks_recently_launched_ORSES_satellite/, n.d.

[28] "Welcome to GNU Radio," GnuRadio wiki site, <http://gnuradio.org/redmine/projects/gnuradio/wiki>, n.d.

[29] Marco Cáceres, "Cost overruns plague military satellite programs," Aerospace America, January 2006, pp. 18-20, 23.

[30] Lyke, James; Young, Quinn; Christensen, Jacob; Anderson, David, "Lessons Learned: Our Decade in Plug-and-play for Spacecraft," Proceedings of the 28th Annual AIAA Conference on Small Satellites, Logan, UT, 2014.

[31] James C. Lyke, "Empowering open systems through cross-platform interoperability," *Proc. SPIE 9096, Open Architecture/Open Business Model Net-Centric Systems and Defense Transformation 2014*, 90960I, June 17, 2014.

[32] Fielding, Roy Thomas, "Architectural Styles and the Design of Network-based Software Architectures," Doctoral dissertation, University of California, Irvine, 2000.
<http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>.

[33] J. Lyke, et al., "SPARC-1: A joint US/Sweden multi-mission modular nanosatellite platform," 2016 IEEE Aerospace Conference, Big Sky, MT, 2016.

[34] Phonebloks project, <http://www.phonebloks.com>, n.d.

[35] "History of Project Ara and modular phones," Project Ara Developers Conference, January 2015, Singapore, <http://modularphonesforum.com>.

[36] Hill, DA; Anderson, LE; Hill, CJ; Mostaghim, A; Rodgers, VGJ; Grover, WH, "Building Blocks for Creating Biological and Chemical Instruments," 2016, *PLoS ONE* 11(7): e0158706. doi:10.1371/journal.pone.0158706.

[37] Space Plug and Play Architecture. AIAA Standards available from <http://arc.aiaa.org/series/aiaastandards>, n.d.

[38] Lyke, J.C., "Plug-and-play satellites," *IEEE Spectrum*, Vol. 49, No. 8, pp. 36-42, August 2012.

[39] J. Lyke, J. Freden, M. Ahlberg, F. Bruhn and J. Preble, "Architectural framework and toolflow concepts for rapidly composable wireless spacecraft," 2016 IEEE Aerospace Conference, Big Sky, MT, 2016.

[40] E. MacDonald, et al., "3D printing for the rapid prototyping of structural electronics," *IEEE Access*, vol. 2, pp. 234–242, Dec. 2014.

[41] Christensen, J., Cannon, S., and B. Hansen, "Automatic Software Generation of ASIM Program Code from an xTEDS," Proceedings of the 2010 AIAA Infotech Conference, 20-22 Apr 2010, Atlanta, GA.

[42] Bilton, Nick, "With Help From Shapeways, You Can Print Your Dishes," *New York Times*, May 12, 2011.

[43] Jobredeaux, Romain; Wang, T.E.; Feron, E.M., "Autocoding control software with proofs I: Annotation translation," in 30th Digital Avionics Systems Conference (DASC), IEEE/AIAA, 16-20 Oct. 2011.

[44] Ramos, A.L.; Ferreira, J.V.; Barcelo, J., "Model-Based Systems Engineering: An Emerging Approach for Modern Systems," *Systems, Man, and Cybernetics, Part C: Applications and Reviews*, *IEEE Transactions*, vol. 42, no. 1, pp.101-111, Jan. 2012.

[45] Kaslow, D.; Anderson, L.; Asundi, S.; Ayres, B.; Iwata, C.; Shiotani, B.; Thompson, R., "Developing a CubeSat Model-Based System Engineering (MBSE) Reference Model - interim status," *Aerospace Conference*, 2015 IEEE, 7-14 March 2015.

[46] H. B. Pandya and T. A. Champaneria, "Internet of things: Survey and case studies," *Electrical, Electronics, Signals, Communication and Optimization (EESCO)*, 2015 IEEE International Conference, Visakhapatnam, India, 2015.

[47] "OmniVision Reaches New Heights; Small, Low-Power CameraChips (TM) Give Spacecraft Sight," *Business Wire*, August 10, 2005, available online: <http://www.businesswire.com/news/home/20050810005214/en/OmniVision-Reaches-Heights-Small-Low-Power-CameraChips-TM>, retrieved January 11, 2015. (no author)

[48] "Prisma," OHB website (online): <http://www.ohb-sweden.se/space-missions/prisma/>, retrieved January 11, 2016.

[49] Nason, I., Puig-Suari, J., and Twiggs, R., "Development of a Family of Picosatellite Deployers Based on the CubeSat Standard," *IEEE Aerospace Conference Proceedings*, Big Sky, MT, 2002.

[50] Barabási, Albert-László, *Linked: The New Science of Networks*, Perseus Books Group, 2002.

[51] Kirk, Jeremy, "Will Megaupload's 28 petabytes of data be deleted?" *Computer World*, March 22, 2012. http://www.computerworld.com/article/2503008/security/0/will-megaupload-s-28-petabytes-of-data-be-deleted_.html.

[52] Rolf Landauer, "Irreversibility and heat generation in the computing process," *IBM Journal of Research and Development*, 5 (3): pp. 183–191, 1961.

LIST OF ACRONYMS, ABBREVIATIONS, AND SYMBOLS

AAIT	Acquire, Assemble, Integrate and Test
AFIT	Air Force Institute of Technology
API	Application Programming Interface
ASIM	applique sensor interface module
ASR	Agile space radio
CCSDS	Consultative Committee for Spacecraft Data Systems
CMOS	Complementary Metal-Oxide Semiconductor
COSMIAC	Configurable Space Microsystems Innovations and Applications Center
CSD	Cannisterized Satellite Dispenser
DHS	Data handling system
EDAC	error detection and correction
ERP	Enterprise Resource Planning
FDIR	Fault Detection Isolation and Recovery
FOI	Swedish Defense Research Agency
FPGA	field programmable gate array
GANT	GPS Antenna
HBNR	High Bandwidth Nanosat Radio
HK	Housekeeping
JAXA	Japanese space agency
LEO	Low Earth Orbit
MOC	Mission operating center
MONARCH	Modular Open Network ARCHitecture
Mbps	millions of bits per second
NAPA	Nanosatellite and plug-and-play architecture
NRZ-L	non return to zero low (protocol)
OBC-S	on-board computer
OIS	Open internet standard
ORS	Operationally Responsive Space
OS	Operating System
PBTF	Push button toolflow
PL1	Payload 1
PL2	Payload 2
PMAD	Power Management and Distribution
POC	Payload operating center
PPOD	poly-picosatellite orbital dispenser
PPS	Pulse Per Second
REST	Representational State Transfer
RF	Radio Frequency
RMAP	remote memory access protocol

RTEMS	Real-time executive for military systems
SPA	Space Plug-and-play Architecture
SPA-1	I2C based SPA standard interface
SPARC	SPA Research Cubesat
SPA-S	spacewire-based SPA standard interface
SpW	spacewire
SSA	Space Situational Awareness
SSM	SPA Services Manager
SSM	SPA Middleware
TCM	telecommand module
TMR	triple modular redundancy
TT&C	timing, telemetry, and control
UART	universal asynchronous receiver transmitter
USN	Universal Space Network
VC	virtual channel
VSI	Virtual System Integration
XML	extensible markup language
xTEDS	XML-based transducer electronic datasheet

DISTRIBUTION LIST

DTIC/OCP
8725 John J. Kingman Rd, Suite 0944
Ft Belvoir, VA 22060-6218 1 cy

AFRL/RVIL
Kirtland AFB, NM 87117-5776 1 cy

Official Record Copy
AFRL/RVSW/James Lyke 1 cy